Tensor techniques for signal processing : algorithms for Canonical Polyadic decomposition

Alex Pereira da Silva

Communauté
UNIVERSITÉ Grenoble Alpes

UNIVERSIDADE FEDERAL DO CEARÁ

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

préparée dans le cadre d'une cotutelle entre la *Communauté Université Grenoble Alpes et l'Université Fédéral du Ceará*

Spécialité : **signal, image, parole, télécommunications (SIPT)**

Arrêté ministériel : le 6 janvier 2005 - 7 août 2006

Présentée par

Alex PEREIRA DA SILVA

Thèse dirigée par **Pierre COMON** et
codirigée par **André L. F. DE ALMEIDA**

préparée au sein du l**aboratoires Grenoble, images, parole, signal, automatique (GIPSA-lab)**

dans **les Écoles doctorales d'électronique, électrotechnique, automatique et traitement du signal (EEATS), et Programa de Pós-Graduação em Engenharia de Teleinformática (PPGETI)**

Techniques tensorielles pour le traitement du signal : algorithmes pour la décomposition polyadique canonique

Thèse soutenue publiquement le **29/06/2016**,
devant le jury composé de :

M. Eric MOREAU
Professeur Université de Toulon, Rapporteur
M. Gérard FAVIER
Directeur de recherche émérite CNRS, Rapporteur
M. Leonardo S. CARDOSO
Maître de conférences INSA-Lyon, Membre
M. Olivier MICHEL
Professeur INP, Président
M. André L. F. DE ALMEIDA
Professeur Université Fédéral du Ceará, Co-Directeur de thèse
M. Pierre COMON
Directeur CNRS, Directeur de thèse

Acknowledgements

I would like to thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico in Brazil for funding this thesis, the director of the GIPSA-lab Jean-Marc Thiriet who gladly received me into the laboratory, and the specialist responsible of the doctorate school EEATS Gang Feng for helping me with important guidance during my PhD progress. I would like to express my gratitude to my thesis advisor, Pierre Comon, to whom I owe my progress as a researcher, for pointing carefully my mistakes and suggesting ideas to develop all this work. I thank my co-advisor André Lima Férrer de Almeida for all fruitful discussions and remarks during my thesis. I could not forget Professor João Cesar who guided me in the beginning of my research still in Brazil.

A special thanks to DECODA team, Souleymen, Yang, Rodrigo, Jeremy, Kostya, Cheng Feng, Miguel, Francesca, for all contributions that they made in the meetings. The discussions about some points of my work were really helpful.

I am grateful to my friends and family to encourage me in hard times, specially my wife, Débora, who was patient and was always by my side supporting me in all circumstances.

Contents

List of Figures

List of Tables

Nomenclature

$\mathbb{K}$	the set of numbers in a generic field.
$\mathbb{N}$	the set of nonnegative integers.
$\mathbb{N}^n$	the set of vectors with n-nonnegative integer entries.
$\mathbb{N}^n_t$	the set of vectors with n-nonnegative integer entries bounded by $t \in \mathbb{N}$ such that $\mathbb{N}^n_t = \{\boldsymbol{\alpha} \in \mathbb{N}^n \| \|\boldsymbol{\alpha}\| := \sum_{i=1}^n \alpha_i \leq t\}$.
$\mathbb{R}^+$	the set of nonnegative real numbers.
$\mathbb{R}[\boldsymbol{x}]$	the ring of multivariate real polynomials in n variables.
$V_{\mathbb{K}}$	$V_{\mathbb{K}}(\mathbb{I})$ is the variety associated to $\mathbb{I} \subseteq \mathbb{R}[\boldsymbol{x}]$, i.e., $V_{\mathbb{K}}(\mathbb{I}) = \{\boldsymbol{x} \in \mathbb{K}^n \| f(\boldsymbol{x}) = 0 \, \forall f \in \mathbb{I}\}$.
$\boldsymbol{x}^{\boldsymbol{\alpha}}$	the monomial whose degree is $\|\boldsymbol{\alpha}\|$, i.e., $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \dots x_n^{\alpha_n}$.
$p_{\boldsymbol{\alpha}}$	the coefficient of the monomial $x^{\boldsymbol{\alpha}}$ of the polynomial $p(\boldsymbol{x})$.
$\succeq$	$\boldsymbol{M} \succeq \mathbf{0}$, means that $\boldsymbol{M}$ is semidefinite positive.
$\lceil \cdot \rceil$	$\lceil x \rceil$ is the smallest integer not less than x.
$\langle \cdot, \cdot \rangle$	the inner product.
$\| \cdot \|$	The Frobenius norm.
$\| \cdot \|_2$	The spectral norm.
$\otimes$	the tensor product.
$\boxtimes$	the Kronecker product.
$\odot$	the Khatri-Rao product.
deg	the degree of a polynomial.
dim	the dimension of a subspace.
Ran	the range or column space. For instance Ran$(\boldsymbol{M})$ is the column space of matrix $\boldsymbol{M}$.
span	the space spanned by a set of vectors,matrices or tensors.
card	the cardinality of a set.
ker	the kernel of a matrix.
trace	the trace of a matrix.
$\mathcal{O}$	Landau's notation for computational complexity. Only multiplications are counted.

$\mathbb{E}\{\cdot\}$	Expected value.
$\delta(x)$	Indicator function defined as: $\delta(x) = 1$ if $x = 0$, and $\delta(x) = 0$ otherwise.
$k_{\boldsymbol{A}}$	Kruskal's rank of the matrix $\boldsymbol{A}$.

Abbreviations and acronyms

ACF *Alternating Conjugated Factors*

ALS *Alternating Least Squares*

CANDCOMP *CANonical DECOMPosition*

CAPD *Combined Alternating Projection and Deflation*

CE *Coupled Eigenvalue*

CP *Canonical Polyadic*

CPS *Conjugated Partially Symmetric*

DS-CDMA *Direct-Sequence Code-Division Multiple Access*

ELS *Enhanced Line Search*

FLOPS *floating-point operations per second*

HOSVD *Higer-Order Singular Value Decomposition*

LHS *Left-Hand Side*

MIMO *Multiple Input - Multiple Output*

PARAFAC *Parallel Factor Analysis*

RHS *Right-Hand Side*

SDP *Semidefinite Programming*

SeROAP *Sequential Rank-One Approximation and Projection*

SOS *Sum-Of-Squares*

ST-HOSVD *Sequentially Truncated HOSVD*

SVD *Singular Value Decomposition*

THOSVD *Truncation of HOSVD*

cte *constant*

s.t. *subject to*

w.r.t. *with respect to*

CHAPTER 1

Introduction

1.1 An introduction to canonical polyadic tensor decomposition

The Canonic Polyadic (CP) tensor decomposition[1] has become an attractive mathematical tool these last ten years in various fields, such as blind source separation [Comon 2010], telecommunications [de Almeida 2007], chemometrics [Smilde 2005], neuroscience [Becker 2014], sensor array processing [Sahnoun 2015] and data mining [Savas 2008]. The interest in resorting to CP tensor decomposition, compared to more standard matrix-based approaches, lies in the uniqueness of the decomposition.

Let $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times \cdots \times I_N}$ be a N-order tensor with entries in some field $\mathbb{K}$, the CP decomposition is given by

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \lambda_r (\boldsymbol{a}_1^{(r)} \otimes \cdots \otimes \boldsymbol{a}_N^{(r)}), \tag{1.1}$$

with $\lambda_r \in \mathbb{K}$, $\boldsymbol{a}_j^{(r)} \in \mathbb{K}^{I_j}$, $1 \leq r \leq R$, $1 \leq j \leq N$, and R is the minimal number of rank-one components, namely, the rank of the tensor [Hitchcock 1927].

Tensors can be represented in a matrix way, which is called *unfoldings*. Actually, there exists several ways to "matricizing" a tensor. Here, one uses the representation in [Kolda 2009]. For an N-order tensor, one has N different unfoldings, each one under each dimension of the tensor. The n-unfolding of a tensor is given by

$$\boldsymbol{T}_{(n)} = \boldsymbol{A}_n \boldsymbol{\Lambda} (\boldsymbol{A}_N \odot \cdots \odot \boldsymbol{A}_{n+1} \odot \boldsymbol{A}_{n-1} \odot \cdots \odot \boldsymbol{A}_1)^{\mathsf{T}}, \tag{1.2}$$

where

$$\boldsymbol{A}_n = \begin{bmatrix} \boldsymbol{a}_n^{(1)} & \cdots & \boldsymbol{a}_n^{(R)} \end{bmatrix} \text{ and } \boldsymbol{\Lambda} = \text{diag}\left(\lambda_1, \cdots, \lambda_R\right).$$

Figure 1.1 illustrates a decomposition of a three-way tensor.

The goal of CP decomposition is to find all components λ_r and $\boldsymbol{a}_i^{(r)}, 1 \leq i \leq N, 1 \leq r \leq R$ from tensor $\boldsymbol{\mathcal{T}}$. However, in practice, tensors are corrupted by noise so that one needs to compute an approximate decomposition since the rank is unknown. The approximation problem is stated as

$$\underset{\lambda_r, \|\boldsymbol{a}_i^{(r)}\|=1}{\arg\min} \; \|\boldsymbol{\mathcal{T}} - \sum_{r=1}^{R} \lambda_r (\boldsymbol{a}_1^{(r)} \otimes \cdots \otimes \boldsymbol{a}_N^{(r)})\|. \tag{1.3}$$

[1] The CP decomposition is also called PARAFAC or CANDECOMP in the literature.

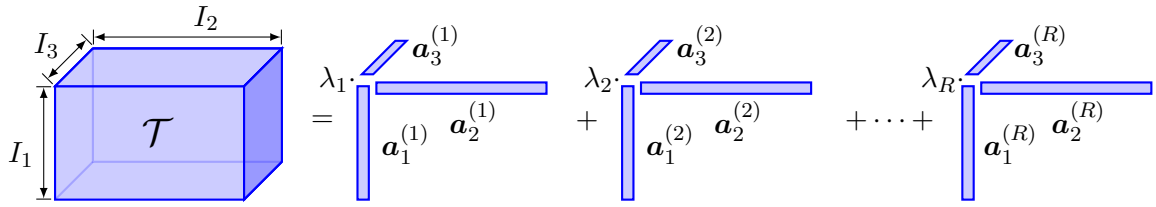


Figure 1.1: CP decomposition of a three-way tensor.

Although finding a low rank approximation is an ill-posed problem in general [De Silva 2008, Hillar 2013], some classical algorithms are still useful to compute CP decomposition. The main workhorse for solving CP is Alternating Least Squares (ALS), which is a simple iterative method that updates alternately the factor matrices $\boldsymbol{A}_n, 1 \leq n \leq N$ in equation 1.2. There are also algorithms based on all-at-once estimation of the matrix factors such as Conjugate Gradient (CG) and Levemberg-Maquardt (LM) methods [Paatero 1997, Comon 2009a]. Some algebraic geometric methods are developed in [Lasserre 2001, Bucero 2014] in order to recover all components, however, there are some limitations with these methods, such as the exponential growth of the number of variables due to convex relaxations which compromises the time of computation for solving the problem. Finally, one cites deflation methods such those in [Cichocki 2009a, Phan 2014], where the main idea is to compute the components of the tensor by applying successive rank-one approximations and subtractions.

In the following, we present the main motivations regarding the choice of the CP decomposition as my research subject, and also the objectives of the thesis.

1.2 Why study the canonical polyadic tensor decomposition?

As mentioned before, the CP decomposition plays a prominent role in a wide number of applications, and it also has advantageous mathematical properties when compared to matrix decompositions. These practical and theoretical points of CP are the main motivations to study this decomposition in my thesis. In the following, we point out the advantage of using tensor approach instead of the matrix one. Next, we draw some example of applications at which the CP decomposition takes place.

Let $\boldsymbol{M}$ be a $I_1 \times I_2$ matrix with entries in some field $\mathbb{K}$. This matrix can be decomposed as a sum of rank-1 matrices as follows

$$\boldsymbol{M} = \sum_{r=1}^{R} \lambda_r \boldsymbol{u}_r \boldsymbol{v}_r^{\mathsf{H}}, \tag{1.4}$$

where $\boldsymbol{u}_r \in \mathbb{K}^{I_1}$ and $\boldsymbol{v}_r \in \mathbb{K}^{I_2}$, $\forall r$. When $\operatorname{rank}(\boldsymbol{M}) = R > 1$, there exist infinitely many factors $\boldsymbol{u}_r, \boldsymbol{v}_r,\ 1 \leq r \leq R$ satisfying the decomposition (1.4). Indeed,

$$\begin{aligned}
\boldsymbol{M} &= \lambda_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathsf{H}} + \lambda_2 \boldsymbol{u}_2 \boldsymbol{v}_2^{\mathsf{H}} + \cdots + \lambda_R \boldsymbol{u}_R \boldsymbol{v}_R^{\mathsf{H}} \\
&= \lambda_1 (\boldsymbol{b} + \boldsymbol{u}_2) \boldsymbol{v}_1^{\mathsf{H}} + \lambda_2 \boldsymbol{u}_2 \boldsymbol{v}_2^{\mathsf{H}} + \cdots + \lambda_R \boldsymbol{u}_R \boldsymbol{v}_R^{\mathsf{H}} \\
&= \lambda_1 \boldsymbol{b} \boldsymbol{v}_1^{\mathsf{H}} + \boldsymbol{u}_2 (\lambda_1 \boldsymbol{v}_1^{\mathsf{H}} + \lambda_2 \boldsymbol{v}_2^{\mathsf{H}}) + \cdots + \lambda_R \boldsymbol{u}_R \boldsymbol{v}_R^{\mathsf{H}},
\end{aligned}$$

for $\boldsymbol{b} = \boldsymbol{u}_1 - \boldsymbol{u}_2 \in \mathbb{K}^{I_1}$. Notice that other choices for $\boldsymbol{b}$ could be made.

In order to ensure a unique decomposition for $\boldsymbol{M}$, we must impose some conditions. Uniqueness is possible, for instance, if the factors $\boldsymbol{u}_r$ and $\boldsymbol{v}_r$, $1 \leq r \leq R$, are orthogonal to one another (i.e. $\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = 0$ and $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0$, $i \neq j$), and $\lambda_i \neq \lambda_j, \forall i \neq j$. These constraints define the singular value decomposition (SVD) of $\boldsymbol{M}$.

The orthogonality of the factors is a strong condition to guarantee the uniqueness of the decomposition of $\boldsymbol{M}$. In fact, if a problem is modeled as in (1.4) but not with orthogonal factors, the estimation of these factors would be very hard or even impossible. Thus, the CP tensor decomposition arises as a powerful tool that ensures the uniqueness of the factors[2] under weaker conditions.

The most known result of uniqueness is a sufficient condition due to Kruskal [Kruskal 1977, Stegeman 2007], which was generalized by other authors to N-th order tensors [Sidiropoulos 2000a]. Let $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_N$ be the factor matrices of a rank-R complex tensor $\boldsymbol{\mathcal{T}}$. If $\sum_{n=1}^{N} k_{\boldsymbol{A}_n} \geq 2R + N - 1$, then the decompostion of $\boldsymbol{\mathcal{T}}$ is essentially unique. $k_{\boldsymbol{A}_n}$ is the Kruskal's rank of the unfolding $\boldsymbol{A}_n$. In addition to the Kruskal's condition, necessary conditions to uniqueness were formulated in [Liu 2001, De Lathauwer 2006]. Besides, the authors in [Lim 2014] present a condition for uniqueness and existence of low rank tensor approximations based on a coherence measure, giving thus an angular interpretation of the approximate decomposition.

Another motivation for studying the CP tensor decompostion is its applicability in several domains. We described here three examples of applications on digital communications [Lim 2014, Sidiropoulos 2000b, de Almeida 2006]. The choice of this domain lies in the fact that part of this thesis was supported by the Department of Teleinformatics Engineering at Federal University of Ceará, so that the interest of developing tensor algorithms keeping in mind this practical domain became central.

Antenna array processing

We show the application in antenna array processing described in [Lim 2014]. Consider a transmission of a wireless communication system composed of n-antenna subarrays, each one with m sensors, and a source in the far field transmitting symbols through a multi-path fading channel. Let $\boldsymbol{b}_i$ be the vector defining the spacial position of the sensors for one of the subarrays called *reference subarray*. The position of the other subarrays with the same sensor structure is defined by the vector $\boldsymbol{\Delta}_j$, $1 \leq j \leq n$. We set $\boldsymbol{\Delta}_1 = \boldsymbol{0}$ since it denotes the reference subarray. Thus, the signal received at discrete time $\boldsymbol{t}_k$, $k = 1, 2, \ldots, l$, on the i-th sensor of subarray j is given by

$$s_{ijk} = \sum_{r=1}^{R} \sigma_r(\boldsymbol{t}_k) \exp(\psi_{ijk}), \tag{1.5}$$

where R is the number of paths, $\sigma_r(\boldsymbol{t}_k)$ is the r-th path of the transmitted signal at time $\boldsymbol{t}_k$, and $\psi_{ijk} = \sqrt{-1}(\omega/c)(\boldsymbol{b}_i^{\mathsf{T}} \boldsymbol{d}_r + \boldsymbol{\Delta}_j^{\mathsf{T}} \boldsymbol{d}_r)$, with ω the pulsation, c the wave celerity and $\boldsymbol{d}_r$ the direction of arrival of the r-th path onto the array structure.

[2] We also use equivalently the term *essentially unique* to indicating that the decomposition is unique with the exception of the elementary indeterminacies of scaling and permutation of factors.

This problem can be formulated as a CP decomposition. Indeed, set $u_{ir} = \exp(\sqrt{-1}(\omega/c)(\boldsymbol{b}_i^{\mathsf{T}}\boldsymbol{d}_r))$, $v_{jr} = \exp(\sqrt{-1}(\omega/c)(\boldsymbol{\Delta}_j^{\mathsf{T}}\boldsymbol{d}_r))$, $w_{kr} = \sigma_r(\boldsymbol{t}_k)/\|\boldsymbol{\sigma}_r\|$, for $\boldsymbol{\sigma}_r = [\sigma_r(\boldsymbol{t}_1)\ \sigma_r(\boldsymbol{t}_2)\ \ldots\ \sigma_r(\boldsymbol{t}_K)]^{\mathsf{T}}$, and $\lambda_r = \|\boldsymbol{\sigma}_r\|$. Thus, we have

$$s_{ijk} = \sum_{r=1}^{R} \lambda_r u_{ir} v_{jr} w_{kr} \iff \boldsymbol{\mathcal{S}} = \sum_{r=1}^{R} \lambda_r \left(\boldsymbol{u}^{(r)} \otimes \boldsymbol{v}^{(r)} \otimes \boldsymbol{w}^{(r)}\right) \in \mathbb{C}^{m\times n\times l}, \tag{1.6}$$

where $\boldsymbol{u}^{(r)} = [u_{1r}\ u_{2r}\ \ldots\ u_{mr}]^{\mathsf{T}}$, $\boldsymbol{v}^{(r)} = [v_{1r}\ v_{2r}\ \ldots\ u_{nr}]^{\mathsf{T}}$, and $\boldsymbol{w}^{(r)} = [w_{1r}\ w_{2r}\ \ldots\ w_{lr}]^{\mathsf{T}}$.

Thereby, solving the CP decomposition provides a blind estimation of the directions $\boldsymbol{d}_r$, the signal wave form $\boldsymbol{\sigma}_r$ and the signal λ_r power, $1 \leq r \leq R$.

DS-CDMA communication

The authors in [Sidiropoulos 2000b] use the CP decomposition to estimate some parameters of a DS-CDMA uplink communication between a single antenna composed of I sensors and R users in a multipath fading baseband channel. Let α_{ir} be the gain of the channel between the sensor i and user r, s_{jr} the j-th symbol transmitted by user r, and c_{kr} the k-th chip of the spreading code of the user r. The baseband output of the i-th sensor of the antenna for symbol period j during chip k can be written as

$$x_{ijk} = \sum_{r=1}^{R} \alpha_{ir} s_{jr} c_{kr} \iff \boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \left(\boldsymbol{\alpha}^{(r)} \otimes \boldsymbol{s}^{(r)} \otimes \boldsymbol{c}^{(r)}\right). \tag{1.7}$$

Hence, the channel gain, the symbols transmitted by the users and the spreading sequence can be estimated by decomposing the received signal $\boldsymbol{\mathcal{X}}$. For more details about interference and synchronization issues of the model see the mentioned article.

Block-fading MIMO channel

Finally, a last example of application in digital communications can be found in [de Almeida 2006]. Therein, the authors represent a block-fading MIMO channel as a CP model. Let B be an $I \times L$ matrix denoting the collection of L path gains during I transmission blocks, $\boldsymbol{A}_R(\theta)$ an $M_R \times L$ matrix whose each element consists of the direction of arrival of the l-th path at the receive antenna m_r, and a matrix $\boldsymbol{C}(\tau, \phi)$ that represents a combined space-time channel response depending on the departure angles ϕ from the transmitted antennas and also on the delay τ of the paths. The dimension of $\boldsymbol{C}(\tau, \phi)$ is $NP \times L$, where N is the size of a training sequence, and P the ratio between the oversampling factor at the receiver and the symbol rate. Thus, the received signal can be represented as a unfolding matrix of a three-way tensor as follows

$$\boldsymbol{X} = \boldsymbol{C}(\tau, \phi)(\boldsymbol{B} \odot \boldsymbol{A}_R(\theta))^{\mathsf{T}}. \tag{1.8}$$

By using an algorithm such as ALS, the channel fading, the arrival and departure angles, and the path delay can be estimated from the CP decomposition.

1.3 Different objectives and the scope of the thesis

This thesis deals with the CP tensor decomposition problems (1.1) and (B.2) in the real and complex fields. Herein, we intend to contribute with new tools to improve the efficiency of finding solutions of these problems considering orthogonal and non-orthogonal tensors. We also present a chapter describing the reduction of quadratic polynomial systems to best rank-1 tensor approximation problems.

In a first stage, our goal is the computation of an approximate CP decomposition when $R = 1$. Since we know that the best rank-1 approximation problem is NP-hard, the idea is to propose suboptimal algorithms that gives good rank-1 approximations. For that, we focus on two types of algorithms: finite and iterative ones. In the first case, our idea was to conceive a simple and efficient algorithm compared to state-of-art ones. Therefore, we propose an algorithm called SeROAP, that has been presenting an excellent performance in terms of approximation error for three-way tensors. In the second case, the goal was the conception of an iterative algorithm presenting some advantage over the standard algorithm ALS. The CE algorithm fulfills this role under some conditions.

In a second stage, our objective is to solve the low rank CP decomposition by using an iterative deflation algorithm similar to the hierarchical method described in [Cichocki 2009c]. However, instead of updating alternately the columns of the factor matrices of a tensor, we perform rank-1 approximations. The reason of proceeding with the computation of rank-1 approximations within the iterative deflation procedure is because we noted that the best rank-1 approximation of a tensor can reduce residues very quickly, ensuring a convergence in a few iterations. Again, the difficult of computing the best rank-1 approximation impels us to use finite rank-1 approximations, such as SeROAP, alternatively, which keeps a reasonable computational complexity as residues are reduced with high probability. It is important to mention that before using an iterative deflation algorithm to compute the CP decomposition, we have performed some theoretical studies to ensure the viability of the method.

Another topic described in my thesis is the application of tensors to solve quadratic polynomial systems. The reduction of a multivariate quadratic system into a best rank-1 approximation problem provides a set of new tools to tackle these kind of systems, and also evinces some advantages compared to other standard mathematical tools, such as those in algebraic geometry and Newton-based methods. Moreover, the best rank-1 approximation always exists and it can be computed by standard algorithms. The reduction is made only in the real field. However, we have got extending the approach to a specific complex system but more general than the real one. In the complex case, we propose an algorithm to compute a constrained rank-1 approximation problem.

Finally, we focus on a particular problem of orthogonal CP tensor decomposition, when one of the factor matrices is semi-unitary. The goal of this part of the manuscript is the proposition of a new algorithm based on the combination of the alternating projection method and the deflation procedure, supported by important convergence guarantees.

1.4 Structure of the thesis and outline

This thesis is composed of a brief introduction, four chapters with results (chapter 2 to 5), and a summary of conclusions and perspectives. We also have included two appendices. The main points of each part of the manuscript is presented in the following.

- *Chapter 1: Introduction.* The formulation of the CP tensor decomposition. The motivations of studying the CP decomposition: theoretical advantages over matrix tools and a wide number of interesting applications. The description of the topics developed during the thesis. The structure of the manuscript and the publications.

- *Chapter 2: Rank-1 tensor approximation.* Formulation of the best rank-1 approximation problem. Description of Lasserre's approach to solve this problem. Description of a finite rank-1 approximation algorithm, called SeROAP, that is always at least as good as the state-of-art finite algorithm called ST-HOSVD. Performance of the finites algorithms. Description of an iterative algorithm, called CE, based on a coupled-eigenvalue problem to compute a rank-1 tensor approximation. Performance of the CE algorithm.

- *Chapter 3: Iterative deflation.* Introduction to the deflation procedure. Description of the iterative deflation algorithm DCPD. Study of the convergence of DCDP under deterministic and probability conditions. Performance of DCDP with finite rank-1 algorithms.

- *Chapter 4: Multivariate quadratic systems and the best rank-1 tensor approximation problem.* Introduction to quadratic polynomial systems. Reduction from quadratic systems to the best rank-1 approximation problem of three-way real tensors. Discussion on the real tensor approach. Extension to the complex field and reduction to a conjugated partially symmetric rank-1 approximation problem. Description of the CPS algorithm. Examples of real and complex systems with/without perturbations. Performance of the tensor approach for solving real and complex generic systems.

- *Chapter 5: Alternating projections on orthogonal CP tensor decomposition.* A brief introduction on alternating projection methods and orthogonal tensors. Description of the CAPD algorithm. Theoretical results on the convergence of CAPD. Results on the orthogonal CP decomposition with a column-wise factor matrix.

1.5 Publications

A. P. da Silva, P. Comon and A. L. F. de Almeida. *An Iterative Deflation Algorithm for Exact CP Tensor Decomposition*. IEEE conference on Acoustics, Speech and Signal Processing, Brisbane, 2015;

A. P. da Silva, P. Comon and A. L. F. de Almeida. *Rank-1 Tensor Approximation Methods and Application to Deflation*. arXiv:1508.05273, 08 2014.

A. P. da Silva, P. Comon and A. L. F. de Almeida. *A Finite Algorithm to Compute Rank-1 Tensor Approximations*. IEEE Signal Processing Letters, 2016.

A. P. da Silva, P. Comon and A. L. F. de Almeida. *On the Reduction of Multivariate Quadratic Systems to Best Rank-1 Approximation of Three-way Tensors*. Applied Mathematics Letters, 2016.

CHAPTER 2

Rank-1 tensor approximation

In this chapter we study alternatives to compute rank-1 tensor approximations in the real and complex fields. We start the chapter with the description of the best rank-1 tensor approximation problem, and the presentation of existing methods to tackle it. The three next sections deal with different approaches to solve rank-1 approximation problems: algebraic geometry, finite methods and iterative methods.

In the algebraic geometry approach, we start by adapting the problem to use Lasserre's relaxation method, which consists of satisfying a constraint condition that ensures the existence of a global solution. Then, we verify if a certificate of optimality is ensured for a set of real tensors with entries distributed according to a probability measure, and an example is drawn up. Additionally, we discuss about the limitations of the method to decompose higher-order and higher-dimension tensors. We conclude the section with a discussion about other relaxation methods.

In the part concerning finite methods, we begin describing three algorithms: T-HOSVD, ST-HOSVD and SeROAP. Therein, we show how these methods work and delineate their main features, including advantages and drawbacks when compared to one another. In the sequel, we give a mathematical proof that our proposed method SeROAP performs at least as good as the other ones in terms of approximation error for three-way tensors. Next, we compare the performance of the algorithms by means of simulations for the following scenarios: three-way tensors and higher-order tensors. The goal is to identify in which configurations SeROAP is useful. The section finishes by comparing the estimation of the rank-1 approximation delivered by the three methods with the best rank-1 approximation obtained by means of Lasserre's method, for $2 \times 2 \times 2$ real tensors.

The part about iterative methods deals with two specified methods: the standard ALS algorithm and the proposed CE algorithm. The goal is to propose for three-way tensors that CE can be a competitive algorithm vis-a-vis the standard ALS. First, we detail how our algorithm is formulated, how it works, and what are its mainly features, in particular the improvement of solutions delivered by other rank-1 methods, and convergence. Second, we evaluate the performance of CE and ALS in terms of the approximation error and the average number of iterations to satisfy a stopping criterion, in order to have a meaningful comparison between both methods.

To close this chapter, we boil down to the main points discussed herein and we also point out some directions to future work.

Main contributions of this chapter:

- *Adaptation of the rank-1 tensor approximation problem to ensure its global solution can be attained by means of Lasserre's method.* We constrain the scalar parameter λ in Problem (B.3) in order to ensure that a specific closed ball be present in the description of the set of constraints,

thereby guaranteeing the global minimum of a relaxed version of this problem (see Appendix (A.2)) is attained;

- *Finite method to compute rank-1 tensor approximation.* We propose an algorithm, called SeROAP, based on successive SVDs and projections, which is less computationally demanding than the T-HOSVD algorithm, and has a competitive complexity compared to ST-HOSVD, for tensors with high dimensions and small order;

- *approximation error of SeROAP algorithm*. We give an analytic proof that the approximation error obtained by using SeROAP is *always* at least as good as T-HOSVD and ST-HOSVD, for three-way tensors;

- *Limitation of SeROAP algorithm*. Heuristics show that SeROAP presents a poorer performance for tensors with order larger than five, when compared to the other algorithms. As a metter of fact, even for 4-th order tensors, ST-HOSVD performs better than SeROAP. However, SeROAP is statistically better than THOSVD with high probability for 4-th order tensors.

- *Iterative rank-1 tensor approximation*. We propose an iterative method, called CE, based on a coupled-eigenvalue problem that computes rank-1 approximations of three-way tensors. The framework of how construct the method is drawn up and some theoretical results ensuring the improvement and convergence of the objective of the rank-1 approximation problem is also shown;

- *Performance of CE method*. The method has competitive advantages compared to the standard ALS. We show that the computational complexity of CE is smaller than ALS when one dimension of the tensor is sufficiently larger than the other ones. Moreover, for two types of initializations, simulations show that CE performs better than ALS in terms of approximation errors and average number of iterations in which a stopping criterion is satisfied.

Contents

2.1 Rank-1 tensor approximation: description and algorithms

Contrary to best rank-r tensor approximations with $r > 1$, the best rank-1 tensor approximation of any tensor always exists in $\mathbb{R}$ or $\mathbb{C}$ since the set of rank-1 tensors is in a cone of Segre varieties [Abo 2009, Landsberg 2012], which ensures its well-posedness. The complexity class of the best rank-1 approximation problem is NP-hard.

An interest in rank-1 approximations lies, for instance, in deflation of symmetric and orthogonally decomposable tensors [Zhang 2001]. Some applications of the best rank-1 approximation are multi-target tracking [Shi 2013], blind source separation [Grellier 2000], and blind equalization [Grellier 1999]. Recently, the authors in [Yang 2016] apply the rank-1 tensor problem to a class of low rank tensor optimization problems. Another interest lies in their use in iterative deflation technique to compute the CP tensor decomposition, which will be explained in Chapter 3.

Let $\lambda \cdot \boldsymbol{a}_1 \otimes \cdots \otimes \boldsymbol{a}_N$ be a rank-1 tensor (or decomposable tensor) in $\mathbb{K}^{I_1 \times I_2 \times \ldots \times I_N}$ with unit factors $\boldsymbol{a}_i, i \in \{1, 2, \ldots, N\}$ and $\lambda \geq 0$. The best rank-1 approximation problem can be stated as

$$\begin{aligned} p^{\min} = \min_{\lambda, \boldsymbol{a}_i \in \mathbb{K}^{I_i}} & \|\boldsymbol{\mathcal{T}} - \lambda \cdot \boldsymbol{a}_1 \otimes \cdots \otimes \boldsymbol{a}_N\|^2 \\ s.t. & \ \|\boldsymbol{a}_i\| = 1. \end{aligned} \tag{2.1}$$

To compute the global minimum of Problem (B.3), we could resort to algebraic geometric tools such as those described in [Lasserre 2001, Bucero 2014]. However, these techniques are only efficiently applied to small-sized real tensors since they introduce a lot of variables due to relaxations, so that computational complexity and storage requirements become an issue.

Still in the algebraic geometry context, [Nie 2014] describes an algorithm that computes best rank-1 approximations in a reasonable time, but the global minimum is attained only if some restrictive rank conditions are satisfied. Again, the drawback is the slow convergence for tensors with moderate dimensions.

Standard iterative algorithms such those described in Appendix A.1 can be employed to tackle Problem (B.3). However, as mentioned in chapter 1, these algorithms do not guarantee that the global minimum will be attained since they are initialization-dependent algorithms, despite the fact that local convergence is ensured for the standard alternating least squares (ALS) algorithm [Uschmajew 2012]. Another specific iterative algorithm to compute the best rank-1 approximation is introduced by Friedland in [Friedland 2013], which is based on the maximal singular pair of matrices but it does not ensure convergence to the global minimum either.

The best rank-1 tensor approximation can also be computed approximately by finite algorithms, i.e., algorithms terminating within a finite number of steps. The solutions obtained by them can be close to the global minimum of Problem (B.3) and they are generally used for initializing standard iterative algorithms. Moreover, due to their low computational complexity, they are also used in the iterative deflation context.

Among existing finite algorithms, we highlight here two algorithms based on the singular value decomposition (SVD). In [De Lathauwer 2000], the authors extend the SVD concept to tensors, and a rank-1 approximation can be obtained by truncating the proposed higher-order singular value decomposition. It is called THOSVD throughout this paper. In [Vannieuwenhoven 2012], a new truncated

strategy to higher order SVD, called ST-HOSVD, presents a computational complexity smaller than that of THOSVD. Moreover, in terms of approximation error, the authors show that ST-HOSVD is always at least as good as THOSVD for three-way real tensors when at least one of the unfolding matrices of the approximating tensor has a rank equal to 1. Therefore, ST-HOSVD has been an efficient alternative to compute a rank-1 three-way tensor approximation [Vannieuwenhoven 2012]. Other finite rank-1 algorithms are also described in [Grellier 1999, Grellier 2000, Hackbusch 2012].

For computing rank-1 approximations, we propose a finite algorithm, called SeROAP, that computes a rank-1 tensor approximation by means of a sequence of singular value decompositions of decreasing order tensors followed by a sequence of projections onto Kronecker vectors of increasing size. The goal of the proposed algorithm is to provide an approximation error at most equal to that obtained with ST-HOSVD (and consequently at most equal to that of THOSVD) for three-way tensors, with computational complexity smaller than that of THOSVD, and still competitive with that of ST-HOSVD, at least for tensors of small orders, regardless of their dimensions. These algorithms are used later in an iterative deflation method to compute low rank tensor approximations.

We also propose an iterative algorithm, called CE, based on an alternating coupled eigenvalue problem. We prove that the objective of Problem B.3 convergences to a limit value. Moreover, under some conditions, the computational complex of CE can be lower than that of the standard ALS algorithm.

2.2 An algebraic geometric approach

The noteworthiness of many tools on algebraic geometry employed in multivariate polynomial optimization brings about the use of these tools to solve rank-1 CP tensor approximation problems. Particularly, polynomial relaxations [Parrilo 2003, Lasserre 2001, Bucero 2014, Laurent 2009] come up with powerful strategies which turn polynomial-based optimization problems into convex problems with approximative solutions.

Moment relaxation [Lasserre 2001] is an attractive method that approximates real-valued polynomial optimization problems over closed semialgebraic sets as closely as desired to a finite sequence of SDP problems. Frequently, this method obtains exactly the global solution of its unrelaxed problem. We give a brief overview of the moment relaxation technique in Appendix A.2. Some variables employed there are used throughout this section.

The rank-1 problem (B.3) can be reformulated as a polynomial form as follows

$$\begin{aligned} p^{\min} = \min_{\lambda\in\mathbb{R}^{+},\boldsymbol{a}_i\in\mathbb{R}^{I_i}} & \|\boldsymbol{\mathcal{T}}_{(n)} - \lambda\cdot\boldsymbol{a}_n(\boldsymbol{a}_N\boxtimes\cdots\boxtimes\boldsymbol{a}_{n+1}\boxtimes\boldsymbol{a}_{n-1}\boxtimes\cdots\boxtimes\boldsymbol{a}_1)^{\mathsf{T}}\|^2 \\ & s.t.\ \|\boldsymbol{a}_i\|^2 = 1, \forall i\in\{1,2,\ldots,N\}, \end{aligned} \tag{2.2}$$

for any unfolding $n\in\{1,2,\ldots,N\}$. The degree of the cost function is reduced to $N+1$ when unit factors $\boldsymbol{a}_i$ are allowed for. Indeed,

$$\begin{aligned} &\|\boldsymbol{\mathcal{T}}_{(n)} - \lambda\cdot\boldsymbol{a}_n(\boldsymbol{a}_N\boxtimes\cdots\boxtimes\boldsymbol{a}_{n+1}\boxtimes\boldsymbol{a}_{n-1}\boxtimes\cdots\boxtimes\boldsymbol{a}_1)^{\mathsf{T}}\|^2 = \\ &\qquad \lambda^2 - 2\lambda\,\mathrm{trace}(\boldsymbol{\mathcal{T}}_{(n)}^{\mathsf{T}}\boldsymbol{a}_n(\boldsymbol{a}_N\boxtimes\cdots\boxtimes\boldsymbol{a}_{n+1}\boxtimes\boldsymbol{a}_{n-1}\boxtimes\cdots\boxtimes\boldsymbol{a}_1)^{\mathsf{T}}) + \|\boldsymbol{\mathcal{T}}\|^2. \end{aligned} \tag{2.3}$$

Problem (2.2) has a unique global minimizer in general [Friedland 2014] up to permutation indeterminacies, which accounts for applying the certificate of optimality pointed in Theorem A.2.2, as far as the moment relaxation approach is concerned. However, in Lasserre's relaxation context, we must ensure that the global minimum of relaxed SDP problems is attained, for instance, in the condition of Proposition A.2.1.

In order to satisfy the condition pointed by Proposition A.2.1, we must add on Problem (2.2) the following ball constraint

$$R^2 - \left(\lambda^2 + \sum_{i=1}^{N} \|\boldsymbol{a}_i\|^2\right) \geq 0, \tag{2.4}$$

where R is the ball radius. The factors $\boldsymbol{a}_i$ are unit vectors, therefore only the variable λ need to be bounded to afford a conformed R. Note that $\lambda \geq 0$ and

$$\|\boldsymbol{\mathcal{T}}_{(n)} - \lambda \cdot \boldsymbol{a}_n(\boldsymbol{a}_N \boxtimes \cdots \boxtimes \boldsymbol{a}_{n+1} \boxtimes \boldsymbol{a}_{n-1} \boxtimes \cdots \boxtimes \boldsymbol{a}_1)^{\mathsf{T}}\|^2 = \|\boldsymbol{\mathcal{T}}\|^2 - \lambda^2,$$

if the Frobenius norm is used. Thus, $\|\boldsymbol{\mathcal{T}}\|^2 - \lambda^2 \geq 0 \implies \lambda \leq \|\boldsymbol{\mathcal{T}}\|$. Therefore,

$$\lambda^2 + \sum_{i=1}^{N} \|\boldsymbol{a}_i\|^2 \leq \|\boldsymbol{\mathcal{T}}\|^2 + N,$$

and we can pick any $R \geq \sqrt{\|\boldsymbol{\mathcal{T}}\|^2 + N}$ to satisfy the inequality (2.4).

Now, we construct the closed semialgebraic set $\mathcal{K}$, defined in Appendix A.2, for Problem (2.2). Note that each equality constraint $\|\boldsymbol{a}_i\|^2 = 1$ can be written as two inequalities: $\|\boldsymbol{a}_i\|^2 - 1 \geq 0$ and $-\|\boldsymbol{a}_i\|^2 + 1 \geq 0$. Thus, we have

$$\begin{aligned}\mathcal{K} = \Big\{ \lambda \in \mathbb{R}, \boldsymbol{a}_i \in \mathbb{R}^{I_i} \big|\ & \lambda \geq 0, \|\boldsymbol{a}_i\|^2 - 1 \geq 0, -\|\boldsymbol{a}_i\|^2 + 1 \geq 0, \\ & \forall i \in \{1, 2, \ldots, N\}, \|\boldsymbol{\mathcal{T}}\|^2 + N - (\lambda^2 + \sum_{i=1}^{N} \|\boldsymbol{a}_i\|^2) \geq 0 \Big\}.\end{aligned} \tag{2.5}$$

We set $R = \sqrt{\|\boldsymbol{\mathcal{T}}\|^2 + N}$. Problem (2.2) is then reformulated as follows

$$p^{\min} = \min_{\mathcal{K}} \|\boldsymbol{\mathcal{T}}_{(n)} - \lambda \cdot \boldsymbol{a}_n(\boldsymbol{a}_N \boxtimes \cdots \boxtimes \boldsymbol{a}_{n+1} \boxtimes \boldsymbol{a}_{n-1} \boxtimes \cdots \boxtimes \boldsymbol{a}_1)^{\mathsf{T}}\|^2. \tag{2.6}$$

The global minimum of its relaxed SDP version, namely p_t^{mom} (as described in Appendix A.2), is thus attained with the optimization over $\mathcal{K}$.

As far as Theoreom A.2.2 is concerned, the certificate of optimality is ensured by simulations, as presented next.

Simulations

In spite of the mathematical elegance of the moment relaxation approach, the size of the involved matrices and the number of variables of the relaxed SDP problems impose a serious drawback. As

a matter of fact, Problem (2.2) has $F = 1 + \sum_{i=1}^{N} I_i$ variables, whereas the relaxed SDP problem in Lasserre's approach has $\binom{F+2t}{2t}$ variables. This stems from the introduction of a new variable for each monomial considering the lexicographic basis [Cox 1992]. Moreover, the size of the moment matrices is $\binom{F+t}{t}$. These limitations enforce computer simulations of small-sized tensors.

Let a sample of 100 real $2 \times 2 \times 2$ tensors whose entries are distributed according to a uniform probability measure in $[-1, 1]$. For this scenario, $d_p = 2$ and $d_{\mathbb{K}} = 1$, so that $t \geq 2$ (vide Appendix A.2). Using the software GloptiPoly© [Henrion 2003, Henrion 2009] we have observed for all tensors that the certificate of optimality is already ensured for $t = 3$, i.e., $p_3^{\text{mom}} = p^{\text{min}}$.

We have selected randomly a tensor from the sample, merely as an example, to verify that the solution obtained by solving the SDP problem for $t = 3$ is the global minimum of Problem (2.2). Let its 1-unfolding

$$\boldsymbol{T}^{(1)} = \begin{bmatrix} 0.9483 & -0.7513 & -0.5911 & -0.2967 \\ -0.9169 & 0.1291 & -0.9392 & -0.2961 \end{bmatrix}.$$

In view of Theorem A.2.2, the condition $\operatorname{rank}(\boldsymbol{M}_2(\boldsymbol{y})) = \operatorname{rank}(\boldsymbol{M}_1(\boldsymbol{y}))$ does not hold for $t = 2$. Indeed, $\operatorname{rank}(\boldsymbol{M}_2(\boldsymbol{y})) = 9$ and $\operatorname{rank}(\boldsymbol{M}_1(\boldsymbol{y})) = 5$. On the other hand, when $t = 3$, the certificate of optimality is ensured inasmuch as $\operatorname{rank}(\boldsymbol{M}_2(\boldsymbol{y})) = \operatorname{rank}(\boldsymbol{M}_1(\boldsymbol{y})) = 4$. GloptiPoly© extracts the best rank-1 approximation whose a global minimizer is

$$\lambda = 1.4791, \boldsymbol{a}_1 = [-0.7701\ 0.6379]^{\mathsf{T}}, \boldsymbol{a}_2 = [0.8944\ -0.4473]^{\mathsf{T}} \text{ and } \boldsymbol{a}_3 = [-0.9951\ -0.0990]^{\mathsf{T}}.$$

The average time of simulations for computing the best rank-1 approximation with moment relaxation approach for our sample was about 23.4826 seconds, which is overwhelming compared to non algebraic approximate algorithms, such as ALS and SeROAP [1]. As a metter of fact, as mentioned before, the size of the relaxed problem increases exponentially. For $2 \times 2 \times 2$ tensors and $t = 2$, the relaxed problem has 330 variables and the size of the moment matrix $\boldsymbol{M}_t(\boldsymbol{y})$ is 36×36. These sizes rise to 1716 and 120×120, respectively, when $t = 3$.

We are not going to show results for larger tensors because the time for computing best rank-1 approximations with GloptiPoly© becomes exponentially slower. In fact, we have ascertained that even simulations in $3 \times 3 \times 3$ scenarios take more than 30 minutes to compute the global solution.

Other relaxation methods

During our research on applying moment relaxation to the best rank-1 approximation problem, we have found out other researches on this topic that were standing out from other ones, such as those in [Nie 2014] and [Bucero 2014].

The former article proposes semidefinite relaxations algorithms based on SOS, that compute best rank-1 approximations if some rank conditions are satisfied, otherwise a nonlinear optimization method should be applied to improve the solution. In spite of these conditions, the authors claim their methods find best rank-1 approximations most of the time. Results for symmetric and non symmetric real tensors are shown even for moderate-sized ones in a reasonable time. For random tensors

[1] The computational time of ALS and SeROAP algorithms is of around few milliseconds for $2 \times 2 \times 2$ tensors.

with dimensions equal to 15 and order 6, their method finds a best rank-1 approximation in only 4 minutes approximately, which would be unfeasible with Laserre's approach. Nonetheless, the time of simulations becomes a drawback for tensors with dimensions larger than 30, even for three-way tensors.

The latter improves the efficiency of Lasserre's moment approach by using border basis reduction [Mourrain 2012]. The authors take the same examples within [Nie 2014] and show that their method computes best rank-1 approximations. Besides, the method is also applied to find the best rank-2 approximation of a symmetric $3 \times 3 \times 3 \times 3$ tensor. The time of simulations is not presented in their results but it is expected to be much faster than Lasserre's.

On account of the auspicious results that had been developing by both aforementioned researches at the time that our research was in progress, we felt ourselves pressured to call off the Lasserre's relaxation approach. As a metter of fact, we have decided to switch to standard tools in linear algebra, as presented in further sections.

2.3 Finite rank-1 approximation algorithms

THOSVD and ST-HOSVD are conceived to compute low multilinear rank approximations. The former was proposed in [De Lathauwer 2000] in the context of the generalization of the SVD. It has been playing a prominent role in many applications where compression is desired, such as signal processing [De Lathauwer 2004, Muti 2005], computer vision [Vasilescu 2002], and data mining [Savas 2003, Savas 2007]. The latter is an alternative method to compute HOSVD with low computational complexity. Contrary to THOSVD, the approximation error computed with ST-HOSVD depends on the order in which the modes are processed, i.e., different order of modes imply different approximation errors. ST-HOSVD is always at least as good as THSOVD for three-way tensors when at least one multilinear rank is equal to 1. Note that the CP rank-1 tensor approximation is equivalent to the multilinear rank-$(1, 1, \ldots, 1)$ approximation, thereby only the rank-1 version of these algorithms are presented in this manuscript.

Our proposed finite method SeROAP computes rank-1 approximations based on a sequence of singular value decompositions followed by a sequence of projections onto Kronecker vectors. The approximation error is at most equal to that of ST-HOSVD for three-way tensors with the same ordering of modes in both algorithms. In this section, we begin describing THOSVD and ST-HOSVD for the rank-1 case and set forth its interest in rank-1 approximations. In a second stage, we draw our attention to SeROAP, describing its formulation and pointing out its advantages.

Truncation of HOSVD

THOSVD is described in the lines of Algorithm 1. Every n-factor is computed by applying the SVD algorithm onto the respective n-mode of $\boldsymbol{\mathcal{T}}$ and keeping the dominant left singular vector $\boldsymbol{u}_n$. With all factors, the rank-1 approximation is obtained by computing the tensor product of the N factors, that is $\boldsymbol{\mathcal{U}}$, and performing the contraction of $\boldsymbol{\mathcal{T}}$ onto $\boldsymbol{\mathcal{U}}$.

We delineate the following points about THSOVD algorithm.

input : $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: input data
output: $\boldsymbol{\mathcal{X}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: rank-1 approximation
for $n = 1$ *to* N **do**
  $\boldsymbol{u}_n \leftarrow$ first left singular vector of $\boldsymbol{T}^{(n)}$;
end
$\boldsymbol{\mathcal{U}} \leftarrow \otimes_{n=1}^{N} \boldsymbol{u}_n$;
$\lambda \leftarrow \langle \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{U}} \rangle$;
$\boldsymbol{\mathcal{X}} \leftarrow \lambda \cdot \boldsymbol{\mathcal{U}}$.

Algorithm 1: THOSVD algorithm - Multilinear rank-$(1, 1, \ldots, 1)$.

- For computing the first left singular vector we do not need to compute the complete SVD in every iteration n, but only the dominant singular values. We use Lanczos algorithm running with a number of k steps [Comon 1990] to compute the singular triplet of a $m \times n$ matrix, with a complexity $\mathcal{O}\{2kmn\}$. In practice, taking k equal to 3 or 4 is enough. Thereby, the accumulated complexity computed for all $\boldsymbol{u}_n$ in THOSVD is equal to $\mathcal{O}\{2Nk \prod_{j=1}^{N} I_j\}$. The computation of $\boldsymbol{\mathcal{U}}$ requires $\prod_{j=1}^{N} I_j$ FLOPS. The contraction to obtain λ (line $\lambda \leftarrow \langle \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{U}} \rangle$) also needs $\prod_{j=1}^{N} I_j$ operations. To sum up, the total number of operations of THOSVD is of order:

$$\mathcal{O}\left\{ (2Nk + 2) \prod_{j=1}^{N} I_j \right\};$$

- Contrary to matrices, the truncated SVD for higher order tensors does not provide a best rank-1 approximation, in spite of keeping a compressed information of $\boldsymbol{\mathcal{T}}$ in multilinear sense. Notice also that in every iteration n, the truncated SVD is calculated for modes of size $\prod_{j=1}^{N} I_j$, which is more computationally expensive when compared with other reduced-order methods (i.e. ST-HOSVD and SeROAP);

Sequentially truncated HOSVD.

The ST-HOSVD algorithm is an alternative strategy that can be used to compute a rank-1 approximation of a tensor [Vannieuwenhoven 2012]. The idea behind this algorithm in this case is to construct the rank-1 tensor approximation with the principal left singular vectors $\boldsymbol{u}_i$, $1 \leq i \leq N$, as in THOSVD, but computed from a sequence of tensors of smaller and smaller order, which in turn are constructed from the principal right singular vectors. Herein, we omit the effect of singular values, present in the description of ST-HOSVD in [Vannieuwenhoven 2012], because they do not affect the computation of the factors for rank-1 approximations. The approximation error $\|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}\|$, for an estimated rank-1 tensor $\boldsymbol{\mathcal{X}}$, depends on the order in which the modes are processed.

Let π_i, $1 \leq i \leq N$, be the unfolding operator applied to a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ along the i-th mode. The unfolding operator is the procedure of reshaping the entries of a tensor into a matrix form [Kolda 2009]. Herein, $\boldsymbol{T}_{(i)} = \pi_i(\boldsymbol{\mathcal{T}})$ is a matrix representation of $\boldsymbol{\mathcal{T}}$ in the matrix space $\mathbb{K}^{I_i \times I_1 I_2 \cdots I_{i-1} I_{i+1} \cdots I_N}$. Let also π_i^{-1}, $1 \leq i \leq N$, be its inverse operator applied to a matrix form that recovers the original tensor $\boldsymbol{\mathcal{T}}$, that is $\boldsymbol{\mathcal{T}} = \pi_i^{-1}(\boldsymbol{T}_{(i)})$. We assume $\boldsymbol{p} = [p_1\ p_2\ \ldots\ p_N]$, with

$p_i \in \{1, 2, \ldots, N\}$ and $p_i \neq p_j$ for $i \neq j$, a vector that defines the ordering of modes processed in ST-HOSVD.

The rank-1 version of ST-HOSVD algorithm is described as follows: the principal left singular vector of the unfolding $\boldsymbol{S}_{(p_1)}$, say $\boldsymbol{u}_{p_1}$, is computed as the first factor of the rank-1 approximation. The tensor $\boldsymbol{\mathcal{S}} = \pi_{p_1}^{-1}(\boldsymbol{v}_{p_1}^{\mathsf{T}})$, where $\boldsymbol{v}_{p_1}$ is the principal right singular vector of $\boldsymbol{S}_{(p_1)}$, is an $N-1$-th order tensor, since $I_{p_1} = 1$. At the next step, $\boldsymbol{u}_{p_2}$ is obtained by computing again the principal left singular vector of the unfolding $\boldsymbol{S}_{(p_2)}$ of $\boldsymbol{\mathcal{S}}$. The tensor $\pi_{p_2}^{-1}(\boldsymbol{v}_{p_2}^{\mathsf{T}})$, where $\boldsymbol{v}_{p_2}$ is the principal right singular vector of $\boldsymbol{S}_{(p_2)}$, is now an $(N-2)$-th order tensor ($I_{p_1} = I_{p_2} = 1$). The process continues according to $\boldsymbol{p}$ until all factors $\boldsymbol{u}_n$ are computed and $\pi_{p_N}^{-1}(\boldsymbol{v}_{p_N}^{\mathsf{T}})$ is a scalar number. In order to obtain a suboptimal solution $\boldsymbol{\mathcal{X}}$, we contract $\boldsymbol{\mathcal{T}}$ onto $\boldsymbol{\mathcal{U}} = \otimes_{n=1}^{N} \boldsymbol{u}_n$, viz., $\lambda = \langle \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{U}} \rangle$, and set the rank-1 approximation as $\boldsymbol{\mathcal{X}} = \lambda \cdot \boldsymbol{\mathcal{U}}$. The method is detailed in Algorithm 2.

input : $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: input data, $\boldsymbol{p}$: processing order
output: $\boldsymbol{\mathcal{X}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: rank-1 approximation
$\boldsymbol{\mathcal{S}} \leftarrow \boldsymbol{\mathcal{T}}$
for $n \leftarrow p_1, p_2, \ldots, p_N$ **do**
  $\boldsymbol{S}_{(n)} \leftarrow \pi_n(\boldsymbol{\mathcal{S}})$
  $\boldsymbol{u}_n \leftarrow$ principal left singular vector of $\boldsymbol{S}_{(n)}$
  $\boldsymbol{v}_n \leftarrow$ principal right singular vector of $\boldsymbol{S}_{(n)}$
  $\boldsymbol{\mathcal{S}} \leftarrow \pi_n^{-1}(\boldsymbol{v}_n^{\mathsf{T}})$
end
$\boldsymbol{\mathcal{U}} \leftarrow \otimes_{n=1}^{N} \boldsymbol{u}_n$;
$\lambda \leftarrow \langle \boldsymbol{\mathcal{T}}, \boldsymbol{\mathcal{U}} \rangle$;
$\boldsymbol{\mathcal{X}} \leftarrow \lambda \cdot \boldsymbol{\mathcal{U}}$.

Algorithm 2: ST-HOSVD algorithm - Multilinear rank-$(1, 1, \ldots, 1)$.

We have some comments about ST-HOSVD algorithm for rank-1 approximations:

- The ordering vector $\boldsymbol{p}$ impacts on the reduction (or increase) of the computational complexity of ST-HOSVD. For computing the principal singular vectors of $\boldsymbol{S}_{(n)}$ at iteration n, we need $\mathcal{O}\{2k \prod_{i=n}^{N} I_{p_i}\}$ operations. Thus, the complexity to compute all factors $\boldsymbol{u}_n$ is of order $\mathcal{O}\{2k \sum_{n=1}^{N-1} \prod_{i=n}^{N} I_{p_i}\}$. Note that if the dimensions of $\boldsymbol{\mathcal{T}}$ were sorted otherwise, the complexity would be different. Moreover, compared to THOSVD, ST-HOSVD has smaller computational complexity. Indeed, as we saw, the main loop of THOSVD has $\mathcal{O}\{2kN \prod_{i=1}^{N} I_i\}$ operations.

- The approximation error of the rank-1 approximation of $\boldsymbol{\mathcal{T}}$ also depends on the choice of the ordering vector $\boldsymbol{p}$. As a metter of fact, for finding the optimal configuration that minimize the approximation error, we should compute the error in $N!$ different configurations, which is intractable for large order tensors;

- Apart from the conditions brought up before, it is not ensured that ST-HOSVD performs as good as the THOSVD algorithm [Vannieuwenhoven 2012].

Sequential Rank-One Approximation and Projection

The proposed algorithm, called SeROAP, is a competitive finite algorithm that computes a rank-1 tensor approximation. As ST-HOSVD, SeROAP also constructs a sequence of tensors of smaller and smaller order, thereby the approximation error depends on the ordering of modes in the process.

Contrary to ST-HOSVD, SeROAP does not compute the factors at every iteration. Instead, the rank-1 approximation is directly computed after a projection process. The order-N version of SeROAP algorithm goes along the lines depicted in Algorithm 3 (see also [da Silva 2014] for a longer description).

input : $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: input data, $\boldsymbol{p}$: processing order
output: $\boldsymbol{\mathcal{X}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: rank-1 approximation
$\boldsymbol{\mathcal{S}} \leftarrow \boldsymbol{\mathcal{T}}$
for $n \leftarrow p_1, p_2, \ldots, p_{N-1}$ **do**
  $\boldsymbol{S}_{(n)} \leftarrow \pi_n(\boldsymbol{\mathcal{S}})$
  $\boldsymbol{v}_n \leftarrow$ principal right singular vector of $\boldsymbol{S}_{(n)}$
  $\boldsymbol{\mathcal{S}} \leftarrow \pi_n^{-1}(\boldsymbol{v}_n^{\mathsf{T}})$
end
$\boldsymbol{u} \leftarrow$ principal left singular vector of $\boldsymbol{S}_{(p_{N-1})}$
$\boldsymbol{w} \leftarrow \boldsymbol{v}_{p_{N-1}}^{*} \boxtimes \boldsymbol{u}$
for $n \leftarrow p_{N-2}, p_{N-3}, \ldots, p_1$ **do**
  $\boldsymbol{X}_{(n)} \leftarrow \boldsymbol{S}_{(n)} \boldsymbol{w} \boldsymbol{w}^{\mathsf{H}}$
  $\boldsymbol{w} \leftarrow \mathrm{vec}(\boldsymbol{X}_{(n)})$
end
$\boldsymbol{\mathcal{X}} \leftarrow \pi_{p_1}^{-1}(\boldsymbol{X}_{(p_1)})$

Algorithm 3: SeROAP algorithm

The SeROAP algorithm can be described in two different phases: decreasing tensor order and projection. The decreasing order phase (first *for* loop) is similar to ST-HOSVD algorithm, except that in SeROAP the factors of the rank-1 approximation are still not computed. As a metter of fact, the goal here is to store the unfoldings $\boldsymbol{S}_{(p_1)}, \boldsymbol{S}_{(p_2)}, \ldots, \boldsymbol{S}_{(p_{N-2})}$ to project their rows into Kronecker vectors that will be obtained in the projection phase. After the first loop, a Kronecker vector $\boldsymbol{w}$ is obtained from the last unfolding $\boldsymbol{S}_{(p_{N-1})}$ by computing its rank-1 approximation. For SeROAP, we do not need to take use of the last element p_N of the mode ordering.

In the projection phase (second *for* loop), at the first step, we project the rows of the matrix $\boldsymbol{S}_{(p_{N-2})}$ onto the Kronecker vector $\boldsymbol{w}$, by performing $\boldsymbol{X}_{(p_{N-2})} = \boldsymbol{S}_{(p_{N-2})} \boldsymbol{w} \boldsymbol{w}^{\mathsf{H}}$. Notice that $\boldsymbol{X}_{(p_{N-2})}$ can be viewed as the unfolding of a rank-1 three-way tensor. Indeed,

$$\begin{aligned} \boldsymbol{X}_{(p_{N-2})} = & \boldsymbol{S}_{(p_{N-2})} \boldsymbol{w} \boldsymbol{w}^{\mathsf{H}} \iff \\ & \boldsymbol{\mathcal{X}}_{(p_{N-2})} = \boldsymbol{S}_{(p_{N-2})} \boldsymbol{w} \otimes \boldsymbol{u}^{*} \otimes \boldsymbol{v}_{p_{N-1}}. \end{aligned}$$

The vector $\boldsymbol{w}$ is updated using the vec operator applied to $\boldsymbol{X}_{(p_{N-2})}$. Since $\boldsymbol{X}_{(p_{N-2})}$ is the unfolding of a three-way tensor, $\mathrm{vec}(\boldsymbol{X}_{(p_{N-2})})$ is a Kronecker product of three vectors: $\mathrm{vec}(\boldsymbol{X}_{(p_{N-2})}) = \boldsymbol{v}_{p_{N-1}} \boxtimes \boldsymbol{u}^{*} \boxtimes \boldsymbol{S}_{(p_{N-2})} \boldsymbol{w}$. Thus, at the next iteration, the unfolding $\boldsymbol{S}_{(p_{N-3})}$ is projected now onto

a *three-way* Kronecker vector. This projection generates a matrix $\boldsymbol{X}_{(p_{N-3})}$ that is the unfolding of a 4-th order tensor. By continuing the process, we note that at the end of the projection phase, the vector $\boldsymbol{w}$ is a Kronecker product of $N-1$ vectors, so that the matrix $\boldsymbol{X}_{(p_1)}$ is actually the unfolding of an order-N rank-1 tensor.

We draw some important aspects about SeROAP method:

- The complexity of SeROAP is $\mathcal{O}\{2k\sum_{n=1}^{N-1}\prod_{i=n}^{N} I_{p_i}\}$ for the decreasing order phase, and $\mathcal{O}\{2\sum_{n=1}^{N-1}\prod_{i=n}^{N} I_{p_i}\}$ for the projection phase. Moreover, all matrices $\boldsymbol{S}_{(p_1)}, \boldsymbol{S}_{(p_2)}, \ldots, \boldsymbol{S}_{(p_{N-2})}$ must be stored before the second phase, which gives a total of $\sum_{n=1}^{N-2}\prod_{i=n}^{N} I_{p_i}$ push operations. Notice that if N is not large, the complexity of SeROAP is smaller than that of THOSVD, and not much larger than that of ST-HOSVD. For instance, for $k=4$, $N=3$ and $I_1=I_2=I_3=100$, we have approximately $24, 8.08,$ and 10.1 million FLOPS for THOSVD, ST-HOSVD and SeROAP algorithms, respectively. For SeROAP, we need 10^6 floating points in memory to store the unfolding matrix $\boldsymbol{S}_{(p_1)}$.
- As ST-HOSVD algorithm, different ordering vectors $\boldsymbol{p}$ yield different approximation errors;
- The idea behind the computation of the factors $\boldsymbol{u}_n$, for $p_1 \leq n < p_N$, from the reshaped form of the right singular vectors $\boldsymbol{v}_n$, lies on the fact that the straight line along $\boldsymbol{v}_n$ fits the row vectors of $\boldsymbol{S}_{(n)}$ such that the sum of squared distances of the row vectors from the line is minimized. Thus, we try to find approximate vectors $\boldsymbol{w}$ fitting the rows of increasing order unfolded tensors $\boldsymbol{\mathcal{X}}_{(n)}$, and thus obtain a good rank-1 approximation;
- For higher order tensors, the successive projections of SeROAP introduces errors in the rank-1 tensor approximation so that the comparison with THOSVD and ST-HOSVD methods could not be recommended. More details in Section 2.3.2.

2.3.1 Theoretical result for rank-1 approximation methods

For three-way tensors, in SeROAP we only need to compute the principal singular triplet of two matrices: one to reduce the order of the tensor, and the other to construct the Kronecker vector. Yet, only a single projection is performed to obtain the rank-1 approximation. In this case, we present in the following a theoretical result showing that the rank-1 approximation computed with SeROAP is *always* at least as good as that delivered by ST-HOSVD, for the same ordering of modes.

Theorem 2.3.1 *Let $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1\times I_2\times I_3}$ be a three-way tensor. $\mathbb{K}=\mathbb{C}$ or $\mathbb{R}$. Let also $\boldsymbol{\mathcal{X}}^{ST}$ and $\boldsymbol{\mathcal{X}}^{Se}$ be the rank-1 approximations delivered by ST-HOSVD and SeROAP algorithms, respectively. Then the approximation error*

$$\|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^{Se}\| \leq \|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^{ST}\|$$

holds for any mode ordering defined by $\boldsymbol{p}$.

Proof: Let $\boldsymbol{p} = [p_1\ p_2\ p_3]$, with $p_i \in \{1,2,3\}$ and $p_i \neq p_j$ for $i \neq j$. Let also $\boldsymbol{T}_{(p_1)}$, $\boldsymbol{X}^{\mathrm{ST}}_{(p_1)}$ and $\boldsymbol{X}^{\mathrm{Se}}_{(p_1)}$ be the unfolding matrix along mode p_1 of tensors $\boldsymbol{\mathcal{T}}$, $\boldsymbol{\mathcal{X}}^{\mathrm{ST}}$ and $\boldsymbol{\mathcal{X}}^{\mathrm{Se}}$, respectively.

Denoting by $\boldsymbol{u}_{p_1}, \boldsymbol{u}_{p_2}$ and $\boldsymbol{u}_{p_3}$ the factors obtained with ST-HOSVD, the approximation error can be written as

$$\|\boldsymbol{\mathcal{T}}_{(p_1)} - \boldsymbol{X}^{\text{ST}}_{(p_1)}\|^2 = \|\boldsymbol{\mathcal{T}}_{(p_1)} - \lambda \boldsymbol{u}_{p_1}(\boldsymbol{u}_{p_3} \boxtimes \boldsymbol{u}_{p_2})^{\mathsf{T}}\|^2,$$

where λ is a positive scalar number for unit factors $\boldsymbol{u}_n, n \in \{p_1, p_2, p_3\}$. Hence,

$$\begin{aligned}\|\boldsymbol{\mathcal{T}}_{(p_1)} - \lambda \boldsymbol{u}_{p_1}(\boldsymbol{u}_{p_3} \boxtimes \boldsymbol{u}_{p_2})^{\mathsf{T}}\|^2 = \\ \|\boldsymbol{\mathcal{T}}\|^2 - \lambda \operatorname{trace}\{\boldsymbol{\mathcal{T}}^{\mathsf{H}}_{(p_1)} \boldsymbol{u}_{p_1}(\boldsymbol{u}_{p_3} \boxtimes \boldsymbol{u}_{p_2})^{\mathsf{T}}\} - \\ - \lambda^* \operatorname{trace}\{\boldsymbol{\mathcal{T}}_{(p_1)}(\boldsymbol{u}^*_{p_3} \boxtimes \boldsymbol{u}^*_{p_2})\boldsymbol{u}^{\mathsf{T}}_{p_1}\} + \lambda^*\lambda. \end{aligned} \tag{2.7}$$

The scalar λ is actually the contraction of $\boldsymbol{\mathcal{T}}$ onto the unit rank-1 tensor $\boldsymbol{u}_{p_1} \otimes \boldsymbol{u}_{p_2} \otimes \boldsymbol{u}_{p_3}$, so that it can be written as

$$\lambda = \operatorname{trace}\{\boldsymbol{\mathcal{T}}^{\mathsf{H}}_{(p_1)} \boldsymbol{u}_{p_1}(\boldsymbol{u}_{p_3} \boxtimes \boldsymbol{u}_{p_2})^{\mathsf{T}}\} = \boldsymbol{u}^{\mathsf{H}}_{p_1} \boldsymbol{\mathcal{T}}_{(p_1)}(\boldsymbol{u}^*_{p_3} \boxtimes \boldsymbol{u}^*_{p_2}). \tag{2.8}$$

Plugging (2.8) into equation (2.7), we obtain, after simplifications,

$$\|\boldsymbol{\mathcal{T}}_{(p_1)} - \boldsymbol{X}^{\text{ST}}_{(p_1)}\|^2 = \|\boldsymbol{\mathcal{T}}\|^2 - |\lambda|^2.$$

Note that $\boldsymbol{u}^{\mathsf{H}}_{p_1} \boldsymbol{\mathcal{T}}_{(p_1)} = \|\boldsymbol{\mathcal{T}}_{(p_1)}\|_2 \, \boldsymbol{v}^{\mathsf{H}}_{p_1}$ for the principal singular triplet $(\boldsymbol{u}_{p_1}, \boldsymbol{v}_{p_1}, \|\boldsymbol{\mathcal{T}}_{(p_1)}\|_2)$ of $\boldsymbol{\mathcal{T}}_{(p_1)}$, where $\|\cdot\|_2$ stands for the spectral norm. Hence

$$\begin{aligned}\lambda = \boldsymbol{u}^{\mathsf{H}}_{p_1} \boldsymbol{\mathcal{T}}_{(p_1)}(\boldsymbol{u}^*_{p_3} \boxtimes \boldsymbol{u}^*_{p_2}) = \|\boldsymbol{\mathcal{T}}_{(p_1)}\|_2 \, \boldsymbol{v}^{H}_{p_1}(\boldsymbol{u}^*_{p_3} \boxtimes \boldsymbol{u}^*_{p_2}) \\ \implies |\lambda|^2 = \|\boldsymbol{\mathcal{T}}_{(p_1)}\|^2_2 \, |\boldsymbol{v}^{\mathsf{H}}_{p_1}(\boldsymbol{u}^*_{p_3} \boxtimes \boldsymbol{u}^*_{p_2})|^2,\end{aligned}$$

which leads to the following approximation error

$$\|\boldsymbol{\mathcal{T}}_{(p_1)} - \boldsymbol{X}^{\text{ST}}_{(p_1)}\|^2 = \|\boldsymbol{\mathcal{T}}\|^2 - \|\boldsymbol{\mathcal{T}}_{(p_1)}\|^2_2 \, |\boldsymbol{v}^{\mathsf{H}}_{p_1}(\boldsymbol{u}^*_{p_3} \boxtimes \boldsymbol{u}^*_{p_2})|^2.$$

On the other hand for SeROAP, we have $\boldsymbol{X}^{\text{Se}}_{(p_1)} = \boldsymbol{\mathcal{T}}_{(p_1)} \boldsymbol{w} \boldsymbol{w}^{\mathsf{H}}$, for $\boldsymbol{w} = \boldsymbol{v}^*_{p_2} \boxtimes \boldsymbol{u}$. Thus,

$$\begin{aligned}\|\boldsymbol{\mathcal{T}}_{(p_1)} - \boldsymbol{X}^{\text{Se}}_{(p_1)}\|^2 &= \|\boldsymbol{\mathcal{T}}_{(p_1)} - \boldsymbol{\mathcal{T}}_{(p_1)} \boldsymbol{w} \boldsymbol{w}^{\mathsf{H}}\|^2 \\ &= \|\boldsymbol{\mathcal{T}}\|^2 + \|\boldsymbol{\mathcal{T}}_{(p_1)} \boldsymbol{w}\|^2 \, \|\boldsymbol{w}\|^2 - 2\|\boldsymbol{\mathcal{T}}_{(p_1)} \boldsymbol{w}\|^2 \\ &= \|\boldsymbol{\mathcal{T}}\|^2 - \boldsymbol{w}^{\mathsf{H}} \boldsymbol{\mathcal{T}}^{\mathsf{H}}_{(p_1)} \boldsymbol{\mathcal{T}}_{(p_1)} \boldsymbol{w}. \end{aligned} \tag{2.9}$$

The vectors $\boldsymbol{u}$ and $\boldsymbol{v}_{p_2}$ are the principal left and right singular vectors of the unfolding $\boldsymbol{S}_{(p_2)}$, as described in Algorithm 3. The eigenvalue decomposition of $\boldsymbol{\mathcal{T}}^{\mathsf{H}}_{(p_1)} \boldsymbol{\mathcal{T}}_{(p_1)}$ can be expressed by

$$\boldsymbol{\mathcal{T}}^{\mathsf{H}}_{(p_1)} \boldsymbol{\mathcal{T}}_{(p_1)} = \|\boldsymbol{\mathcal{T}}_{(p_1)}\|^2_2 \boldsymbol{v}_{p_1} \boldsymbol{v}^{\mathsf{H}}_{p_1} + \boldsymbol{V}, \tag{2.10}$$

where $\boldsymbol{V}$ is a semidefinite positive matrix. Plugging (2.10) into (2.9)

$$\begin{aligned}\|\boldsymbol{\mathcal{T}}_{(p_1)} - \boldsymbol{X}^{\text{Se}}_{(p_1)}\|^2 &= \|\boldsymbol{\mathcal{T}}\|^2 - \|\boldsymbol{\mathcal{T}}_{(p_1)}\|^2_2 \boldsymbol{w}^{\mathsf{H}} \boldsymbol{v}_{p_1} \boldsymbol{v}^{\mathsf{H}}_{p_1} \boldsymbol{w} - c \\ &= \|\boldsymbol{\mathcal{T}}\|^2 - \|\boldsymbol{\mathcal{T}}_{(p_1)}\|^2_2 |\boldsymbol{v}^{\mathsf{H}}_{p_1} \boldsymbol{w}|^2 - c,\end{aligned}$$

with $c = \boldsymbol{w}^{\mathsf{H}}\boldsymbol{V}\boldsymbol{w} \geq 0$.
To complete the proof of the theorem, we just need to show that $|\boldsymbol{v}_{p_1}^{\mathsf{H}}\boldsymbol{w}|^2 \geq |\boldsymbol{v}_{p_1}^{\mathsf{H}}(\boldsymbol{u}_{p_3}^* \boxtimes \boldsymbol{u}_{p_2}^*)|^2$, or equivalently that

$$|\langle \boldsymbol{w}, \boldsymbol{v}_{p_1} \rangle| \geq |\langle \boldsymbol{u}_{p_3}^* \boxtimes \boldsymbol{u}_{p_2}^*, \boldsymbol{v}_{p_1} \rangle|.$$

This is true, because $\boldsymbol{w}$ is by construction (cf. Algorithm 3) the vector closest to $\boldsymbol{v}_{p_1}$ among all vectors of the form $\boldsymbol{a} \boxtimes \boldsymbol{b}$ where $\boldsymbol{a}$ and $\boldsymbol{b}$ have unit norm. □

Corollary 2.3.2 *Let $\boldsymbol{\mathcal{X}}^{TH}$ be the rank-1 approximation delivered by THOSVD algorithm. Then the approximation error*

$$\|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^{Se}\| \leq \|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^{TH}\|$$

holds for any mode ordering defined by $\boldsymbol{p}$.

Proof: The proof follows directly from Theorem 2.3.1 above and from Theorem 7.2 described in [Vannieuwenhoven 2012], in which the approximation error computed with ST-HOSVD is at most equal to that obtained with THOSVD for three-way tensors, when at least one of the multilinear ranks of the approximating tensor is equal to 1. □

2.3.2 Performance of finite rank-1 approximation methods

In this section, we are going to evaluate the performance of SeROAP, THOSVD and ST-HOSVD in terms of approximation error. Firstly, we compare SeROAP and THOSVD for different three-way tensor sizes with entries in the complex field, and considering three different order of modes. The same comparison is made in the following for SeROAP and ST-HOSVD. Secondly, we evaluate all methods under higher order tensor scenarios. Finally, we measure how far the estimated rank-1 approximations are from the best rank-1 approximation, which is obtained from Lasserre's approach.

Scenarios of three-way tensors

To compare the approximation error of the rank-1 approximation methods SeROAP, THOSVD and ST-HOSVD, we have simulated four tensor scenarios: $3 \times 4 \times 5$, $3 \times 4 \times 20$, $3 \times 20 \times 20$ and $20 \times 20 \times 20$, which are labeled as scenarios 1, 2, 3 and 4, respectively. With these scenarios, we have intended to take into account cases with equal dimensions or not. For each of them, a sample of 300 complex tensors with real and imaginary parts uniformly distributed in $[-1, 1]$ was generated thereby ensuring a coherent and meaningful comparison.

As mentioned in previous sections, the approximation error of SeROAP depends on the choice of the order of modes. Therefore, we have computed the solution delivered by SeROAP in three different modes defined by the permutations $\boldsymbol{p}_1 = [1\ 2\ 3]$, $\boldsymbol{p}_2 = [2\ 3\ 1]$, and $\boldsymbol{p}_3 = [3\ 1\ 2]$.

In order to characterize the comparison between SeROAP and THOSVD methods, we introduce the following metric

$$\Delta\phi_t = \|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^{\mathrm{TH}}\| - \|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^{\mathrm{Se}}\|,$$

which is the difference between the approximation errors obtained by both methods. Notice that $\Delta\phi_t > 0$ implies that SeROAP performs better than THOSVD.

Figures 2.1, 2.2 and 2.3 depict the better performance of SeROAP over THSOVD for $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, and $\boldsymbol{p}_3$, respectively.

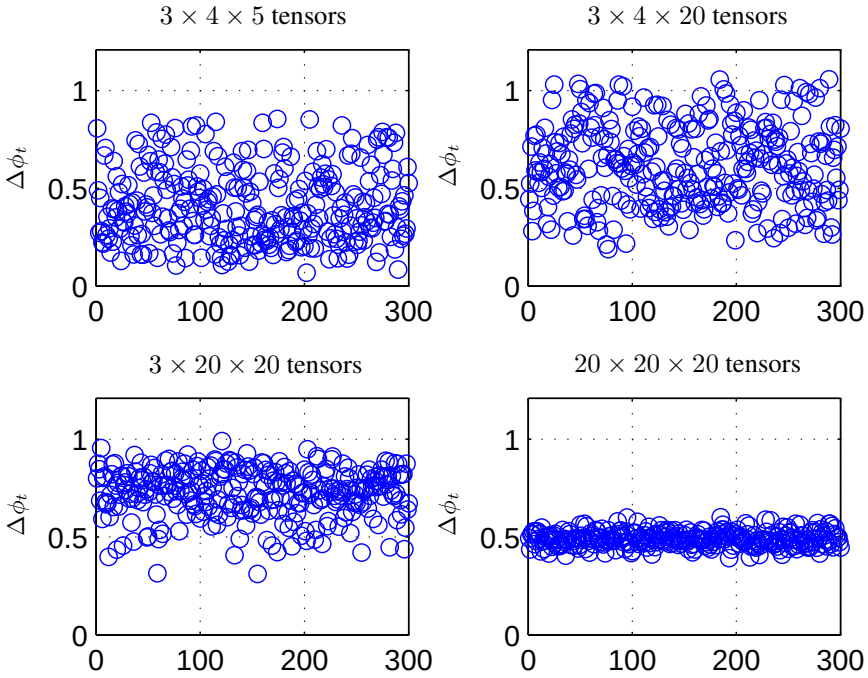


Figure 2.1: SeROAP vs THOSVD - $\boldsymbol{p}_1 = [1\ 2\ 3]$.

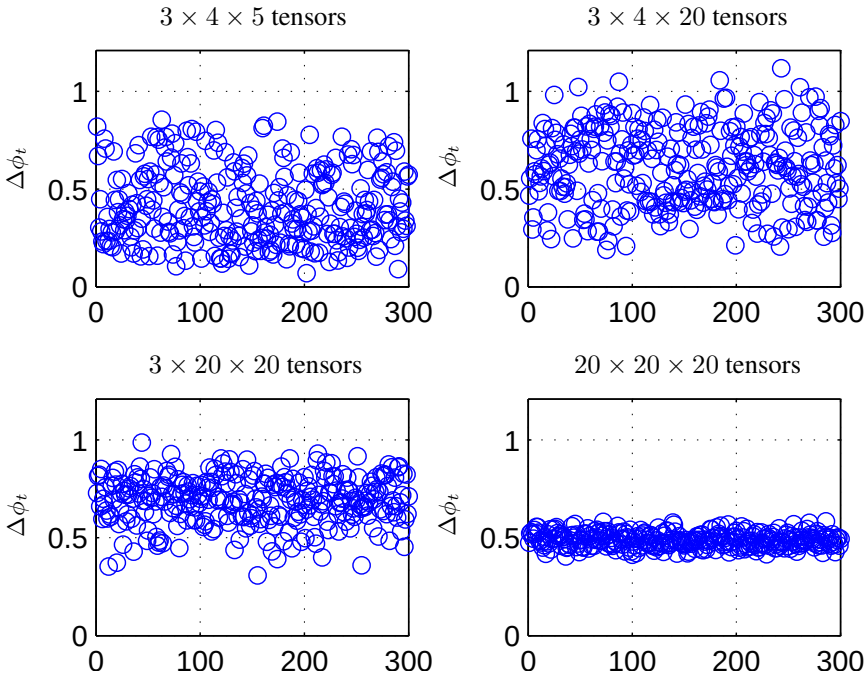


Figure 2.2: SeROAP vs THOSVD - $\boldsymbol{p}_2 = [2\ 3\ 1]$.

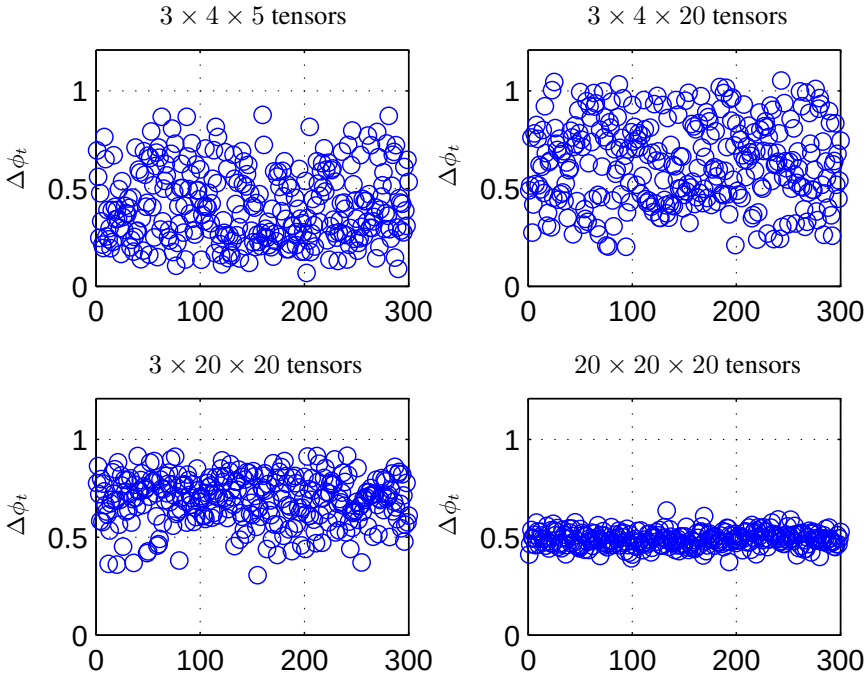


Figure 2.3: SeROAP vs THOSVD - $\boldsymbol{p}_3 = [3\ 1\ 2]$.

As predicted by Proposition 2.3.1, the results show that the approximation error is smaller for SeROAP algorithm. Indeed, it turns out that $\Delta\phi_t > 0$ for the three order of modes employed in

SeROAP.

The behavior of $\Delta\phi_t$ does not change significantly throughout the different permutations. However, with the variation of dimension sizes, we have noted different distribution of $\Delta\phi_t$ values. As a metter of fact, the larger the dimensions, more $\Delta\phi_t$ concentrates around some value. For instance, the scenario with $20 \times 20 \times 20$ tensors, $\Delta\phi_t \approx 0.5$. Notice also the distribution of $\Delta\phi_t$ values reduces as the size of dimensions increases (compare scenarios 2, 3 and 4).

Now, we focus on the comparison of the approximation error obtained with SeROAP and ST-HOSVD. As defined before, we use the following metric that computes the difference between the approximation error evaluated by both algorithms.

$$\Delta\phi_s = \|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^{\mathrm{ST}}\| - \|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^{\mathrm{Se}}\|,$$

Again, if $\Delta\phi_s > 0$, SeROAP algorithm performs better than ST-HOSVD in terms of approximation error.

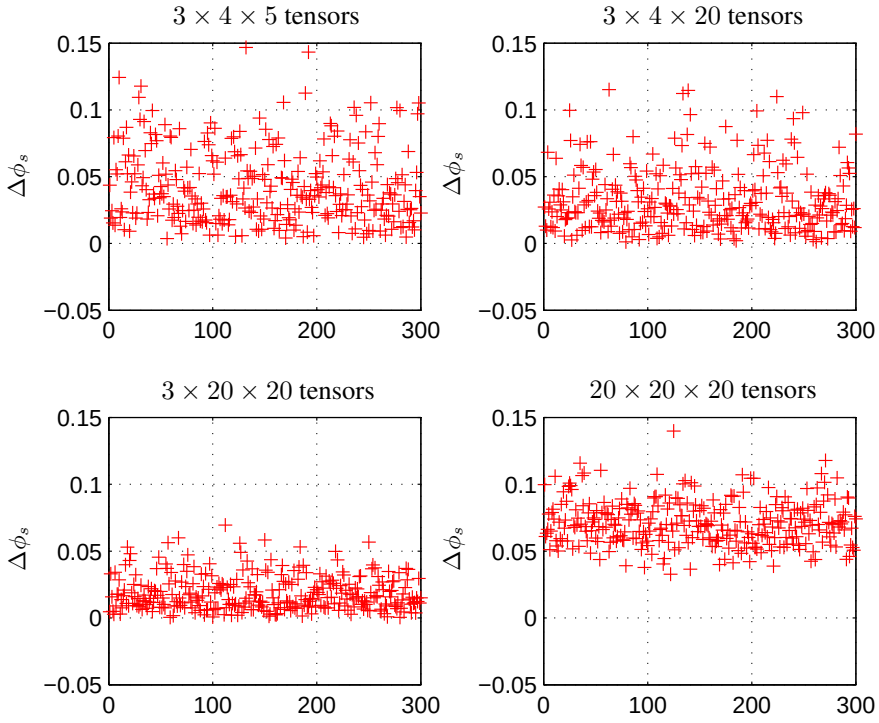


Figure 2.4: SeROAP vs ST-HOSVD - $\boldsymbol{p}_1 = [1\ 2\ 3]$.

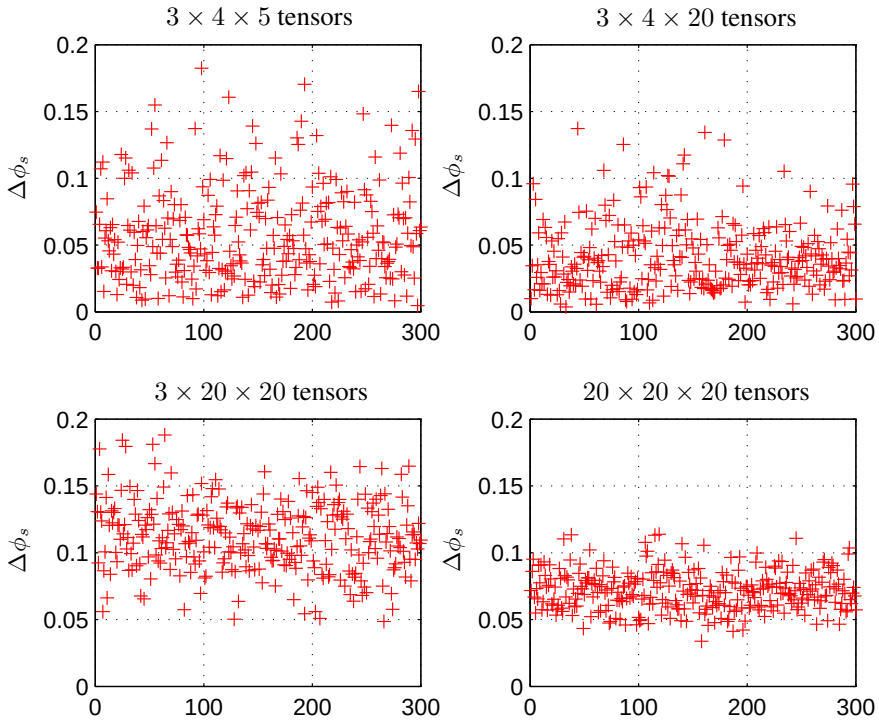


Figure 2.5: SeROAP vs ST-HOSVD - $\boldsymbol{p}_2 = [2\ 3\ 1]$.

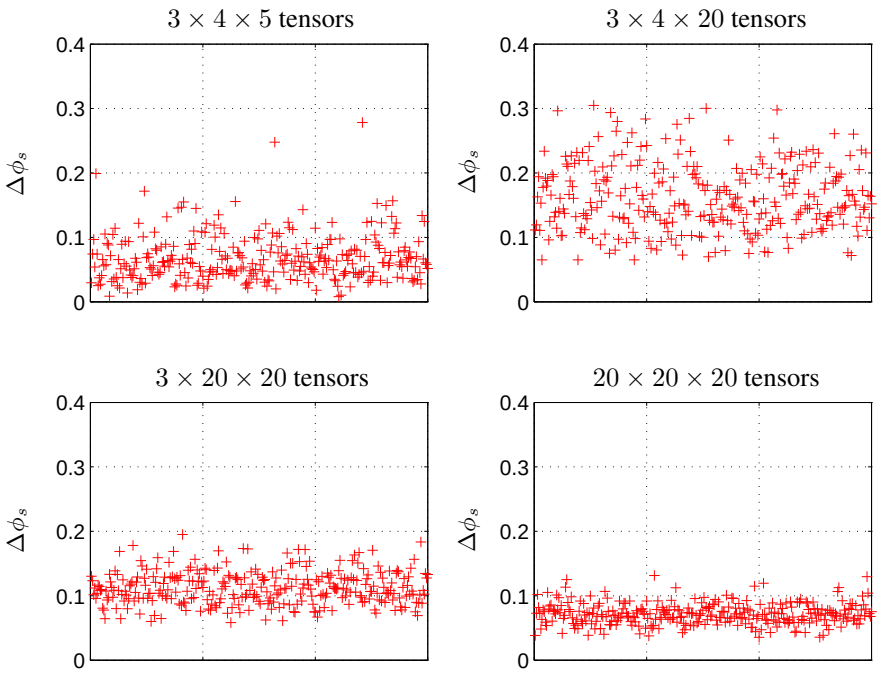


Figure 2.6: SeROAP vs ST-HOSVD - $\boldsymbol{p}_3 = [3\ 1\ 2]$.

Contrary to THOSVD algorithm, the behavior of the approximation error depends on the order of modes, which can be viewed in our results. Note that for $\boldsymbol{p}_1$, the values of $\Delta\phi_s$ in the scenario $3 \times 20 \times 20$ are close to zero, which implies that SeROAP and ST-HOSVD have approximately the same approximation errors. Indeed, for this scenario, $0 \leq \Delta\phi_s \leq 0.05$ for $97.33\ \%$ of tensors.

When the ordering vector used in both algorithms is $\boldsymbol{p}_2$ or $\boldsymbol{p}_3$, SeROAP provides a better performance compared to ST-HOSVD. This can be noted throughout the three last scenarios inasmuch as $\Delta\phi_s$ values are rather far from zero. As expected, results also confirm Theorem 2.3.1 irrespective the employed scenarios.

It is important to mention that our results reinforce the result described in [Vannieuwenhoven 2012] in which ST-HOSVD outpeforms THOSVD in terms of approximation error. Indeed, while $\Delta\phi_t$ varies from 0.1 and 1.1, the values of $\Delta\phi_s$ fluctuates between 0 and 0.4, approximately.

Scenarios of higher order tensors

Theorem 2.3.1 ensures that the error of the rank-1 approximation of SeROAP is smaler than those of THSOVD and ST-HOSVD methods only for three-way tensors. In this section, we evaluate the impact on the approximation errors when the order of the tensors is higher than 3.

As before, we consider a sample of 300 complex tensors with real and imaginary parts uniformly distributed in $[-1, 1]$. We evaluate two higher-order scenarios: $3\times^4$ and $3\times^5$. We have chosen for both algorithms the ordering $\boldsymbol{p}_1 = [1\ 2\ 3\ 4]$ and $\boldsymbol{p}_2 = [1\ 2\ 3\ 4\ 5]$. The metrics $\Delta\phi_t$ and $\Delta\phi_s$ are depicted in Figure 2.7.

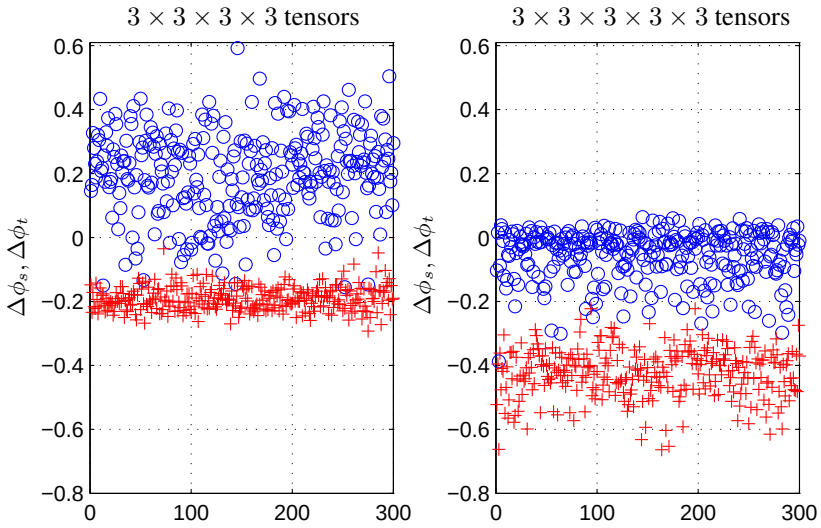


Figure 2.7: SeROAP vs ST-HOSVD and THOSVD - 4-th and 5-th order tensors.

For both scenarios, we note that Theorem 2.3.1 does not hold anymore. Indeed, for the 4-th order tensor scenario, it turns out by our numerical experiments that SeROAP delivers a better rank-1 approximation with high probability than that of THOSVD, but its performance worsens drastically when compared to ST-HOSVD. The approximation errors computed with SeROAP is even worst for

5-th order tensors. Since the successive projections of SeROAP introduces substantial errors in the rank-1 tensor approximation, the comparison is not recommended for higher order tensors.

Comparison to best rank-1 approximation in $2 \times 2 \times 2$ scenario

Now, we assess the approximation errors in $2 \times 2 \times 2$ scenario of the previous finite methods by comparing them to the best rank-1 approximation, which is computed using Lasserre's approach. We have generated a sample of 200 real tensors with entries uniformly distributed in $[-1, 1]$. In simulations, we have used only the permutation vector $\boldsymbol{p} = [1\ 2\ 3]$. In order to have a meaningful comparison, we consider the MSE metric given by

$$\text{MSE} = \frac{1}{200} \sum_{n=1}^{200} \left(\Delta\phi_m^{[n]} \right)^2,$$

where $\Delta\phi_m^{[n]} = \|\boldsymbol{\mathcal{T}}^{[n]} - \boldsymbol{\mathcal{X}}_m^{[n]}\| - \|\boldsymbol{\mathcal{T}}^{[n]} - \boldsymbol{\mathcal{X}}^{[n]\star}\|$. Here, $m \in \{\text{“}TH\text{”}, \text{“}ST\text{”}, \text{“}Se\text{”}\}$ is a tag indicating the method, $\boldsymbol{\mathcal{X}}_m^{[n]}$ is the rank-1 approximation of $\boldsymbol{\mathcal{T}}^{[n]}$ delivered by the method m, and $\boldsymbol{\mathcal{X}}^{[n]\star}$ is the best rank-1 approximation of $\boldsymbol{\mathcal{T}}^{[n]}$. Clearly, $\Delta\phi_m^{[n]} \geq 0,\ \forall m, n$.

We also point out the percentage of tensors in which $\Delta\phi_m^{[n]} \leq 10^{-4}$ in order to measure how close to the best approximation the solutions of the finite methods are. The table bellow depicts these results.

Algorithm	m	MSE	$\%\ \Delta\phi_m^{[n]} \leq 10^{-4}$
THOSVD	“TH"	0.0230	1.5 %
ST-HOSVD	“ST"	0.0013	11.0 %
SeROAP	“Se"	2.4511e-04	26.5 %

Table 2.1: Performance of finite rank-1 methods for $2 \times 2 \times 2$ tensors.

As expected, SeROAP exhibits the lowest MSE among all methods, and THOSVD the worst one. Moreover, SeROAP also delivers rank-1 approximations close to the best one more often than ST-HOSVD and THOSVD. While only for 3 tensors the condition $\Delta\phi_m^{[n]} \leq 10^{-4}$ hold s in THSOVD, a total of 53 tensors have satisfied this condition in SeROAP. ST-HOSVD has got the second better performance.

2.4 Iterative rank-1 approximation methods

To compute rank-1 approximations using iterative methods, we can use those ones mentioned in the Introduction. Due to high complexity of Newton and gradient-based methods, ALS algorithm is usually preferred to do this task. As a metter of fact, it is easy to be implemented and ensure satisfactory results in general. Additionally, ALS also presents a general global convergence property for rank-1 approximations [Wang 2014]. Indeed, for a given initialization point, a sequence of iterates always converge to a unique limit point in general.

To compete with ALS, it is necessary to propose some method that combines simple implementability and good convergence features. For this, we propose a new method for three-way tensors, called CE (Coupled-Eigenvalue), that:

- presents less computational complexity than ALS when one dimension is larger enough than the other ones;
- for any initialization, the approximation error always converges to a limit value;
- performs better than ALS in terms of approximation error and convergence rate in most cases according to simulations;

2.4.1 Coupled-eigenvalue rank-1 approximation

This section presents an alternating eigenvalue method for three-way tensors that can improve local solutions obtained from any other rank-1 approximation method (e.g. SeROAP, ST-HOSVD and THOSVD algorithms).

Let $\boldsymbol{\mathcal{T}}$ be a tensor in $\mathbb{K}^{I_1 \times I_2 \times I_3}$ and $\boldsymbol{t}_{i_n}$ be the vectorization of its slice i_n, $1 \leq i_n \leq I_n$, and $n \in \{1, 2, 3\}$. Without loss of generality, we assume $n = 3$. The rank-1 approximation problem B.3 can be stated as

$$\begin{aligned} [\boldsymbol{z}_{opt}, \boldsymbol{x}_{opt}, \boldsymbol{y}_{opt}] &= \arg\min_{\boldsymbol{z},\boldsymbol{x},\boldsymbol{y}} \Upsilon(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \\ s.t.\ \|\boldsymbol{x}\| &= 1,\ \|\boldsymbol{y}\| = 1, \end{aligned} \tag{2.11}$$

with $\Upsilon(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = \sum_{i_3=1}^{I_3} \|\boldsymbol{t}_{i_3} - z_{i_3}(\boldsymbol{y} \boxtimes \boldsymbol{x})\|^2$ and $\boldsymbol{z} = [z_1 \cdots z_{I_3}]$. Indeed,

$$\|\boldsymbol{\mathcal{T}} - \boldsymbol{x} \otimes \boldsymbol{y} \otimes \boldsymbol{z}\|^2 = \|\boldsymbol{T}^{(3)} - \boldsymbol{z}(\boldsymbol{y} \boxtimes \boldsymbol{x})^{\mathsf{T}}\|^2 = \sum_{i_3=1}^{I_3} \|\boldsymbol{t}_{i_3} - z_{i_3}(\boldsymbol{y} \boxtimes \boldsymbol{x})\|^2.$$

Let $\mathcal{L}_0$ be the lagrangian of Problem (2.11) given by

$$\mathcal{L}_0(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \eta_1, \eta_2) = \sum_{i_3=1}^{I_3} \|\boldsymbol{t}_{i_3} - z_{i_3}(\boldsymbol{y} \boxtimes \boldsymbol{x})\|^2 + \eta_1(\|\boldsymbol{y}\|^2 - 1) + \eta_2(\|\boldsymbol{x}\|^2 - 1),$$

where η_1 and η_2 are the Lagrange multipliers.

By computing the stationary points of $\mathcal{L}_0$, we obtain for each z_{i_3} the following

$$\frac{\partial \mathcal{L}_0}{\partial z_{i_3}^*} = -(\boldsymbol{y} \boxtimes \boldsymbol{x})^{\mathsf{H}} \boldsymbol{t}_{i_3} + z_{i_3} = 0 \implies z_{i_3} = (\boldsymbol{y} \boxtimes \boldsymbol{x})^{\mathsf{H}} \boldsymbol{t}_{i_3}.$$

Plugging the value of z_{i_3} into Problem (2.11), we can rewrite it as the following equivalent maximization problem

$$\begin{aligned} [\boldsymbol{x}_{opt}, \boldsymbol{y}_{opt}] &= \arg\max_{\boldsymbol{x},\boldsymbol{y}} (\boldsymbol{y} \boxtimes \boldsymbol{x})^{\mathsf{H}} \boldsymbol{M} (\boldsymbol{y} \boxtimes \boldsymbol{x}) \\ s.t.\ \|\boldsymbol{y}\|_2 &= 1, \|\boldsymbol{x}\|_2 = 1, \end{aligned} \tag{2.12}$$

where $\boldsymbol{M} = \sum_{i_3=1}^{I_3} \boldsymbol{t}_{i_3} \boldsymbol{t}_{i_3}^{\mathsf{H}}$.

Now, we decompose $\boldsymbol{M}$ as a sum of Kronecker products. This can be done by reshaping $\boldsymbol{M}$ and applying the SVD decomposition [Van Loan 1993]. Thus, $\boldsymbol{M}$ can be given by

$$\boldsymbol{M} = \sum_{r=1}^{R'} \boldsymbol{Q}^{(r)} \boxtimes \boldsymbol{P}^{(r)},$$

with matrices $\boldsymbol{P}^{(r)} \in \mathbb{K}^{I_1 \times I_1}$ and $\boldsymbol{Q}^{(r)} \in \mathbb{K}^{I_2 \times I_2}$. R' is the *Kronecker rank* of $\boldsymbol{M}$ satisfying $R' \leq I_1 I_2$. Substituting $\boldsymbol{M}$ into Problem (2.12), we have:

$$\begin{aligned} [\boldsymbol{x}_{opt}, \boldsymbol{y}_{opt}] &= \arg\max_{\boldsymbol{x},\boldsymbol{y}} \Gamma(\boldsymbol{x}, \boldsymbol{y}) \\ s.t. &\|\boldsymbol{y}\|_2 = 1,\ \|\boldsymbol{x}\|_2 = 1, \end{aligned} \tag{2.13}$$

with $\Gamma(\boldsymbol{x}, \boldsymbol{y}) = \sum_{r=1}^{R'} (\boldsymbol{y}^{\mathsf{H}} \boldsymbol{Q}^{(r)} \boldsymbol{y})(\boldsymbol{x}^{\mathsf{H}} \boldsymbol{P}^{(r)} \boldsymbol{x})$. This holds because

$$(\boldsymbol{y} \boxtimes \boldsymbol{x})^{\mathsf{H}} \left(\sum_{r=1}^{R'} \boldsymbol{Q}^{(r)} \boxtimes \boldsymbol{P}^{(r)} \right) (\boldsymbol{y} \boxtimes \boldsymbol{x}) = \sum_{r=1}^{R'} (\boldsymbol{y} \boxtimes \boldsymbol{x})^{\mathsf{H}} (\boldsymbol{Q}^{(r)} \boxtimes \boldsymbol{P}^{(r)})(\boldsymbol{y} \boxtimes \boldsymbol{x}) =$$

$$= \sum_{r=1}^{R'} (\boldsymbol{y}^{\mathsf{H}} \boldsymbol{Q}^{(r)} \boxtimes \boldsymbol{x}^{\mathsf{H}} \boldsymbol{P}^{(r)})(\boldsymbol{y} \boxtimes \boldsymbol{x}) = \sum_{r=1}^{R'} (\boldsymbol{y}^{\mathsf{H}} \boldsymbol{Q}^{(r)} \boldsymbol{y}) \boxtimes (\boldsymbol{x}^{\mathsf{H}} \boldsymbol{P}^{(r)} \boldsymbol{x}) = \sum_{r=1}^{R'} (\boldsymbol{y}^{\mathsf{H}} \boldsymbol{Q}^{(r)} \boldsymbol{y})(\boldsymbol{x}^{\mathsf{H}} \boldsymbol{P}^{(r)} \boldsymbol{x}).$$

Now, let $\mathcal{L}$ be the Lagrangian of Problem (2.13)given by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \eta_3, \eta_4) = -\Gamma(\boldsymbol{x}, \boldsymbol{y}) + \eta_3(\|\boldsymbol{y}\|_2^2 - 1) + \eta_4(\|\boldsymbol{x}\|_2^2 - 1).$$

By computing the critical points, we obtain a pair of coupled eigenvalue problems

$$\begin{bmatrix} \boldsymbol{y}^{\mathsf{H}} \boldsymbol{A}_{(1,1)} \boldsymbol{y} & \cdots & \boldsymbol{y}^{\mathsf{H}} \boldsymbol{A}_{(1,I_1)} \boldsymbol{y} \\ \vdots & \ddots & \vdots \\ \boldsymbol{y}^{\mathsf{H}} \boldsymbol{A}_{(I_1,1)} \boldsymbol{y} & \cdots & \boldsymbol{y}^{\mathsf{H}} \boldsymbol{A}_{(I_1,I_1)} \boldsymbol{y} \end{bmatrix} \boldsymbol{x} = \lambda \boldsymbol{x} \tag{2.14}$$

and

$$\begin{bmatrix} \boldsymbol{x}^{\mathsf{H}} \boldsymbol{B}_{(1,1)} \boldsymbol{x} & \cdots & \boldsymbol{x}^{\mathsf{H}} \boldsymbol{B}_{(1,I_2)} \boldsymbol{x} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x}^{\mathsf{H}} \boldsymbol{B}_{(I_2,1)} \boldsymbol{x} & \cdots & \boldsymbol{x}^{\mathsf{H}} \boldsymbol{B}_{(I_2,I_2)} \boldsymbol{x} \end{bmatrix} \boldsymbol{y} = \lambda \boldsymbol{y}, \tag{2.15}$$

where $\lambda = \eta_3 = \eta_4$, $\boldsymbol{A}_{(m,n)} = \sum_{r=1}^{R'} P_{mn}^{(r)} \boldsymbol{Q}^{(r)}$, and $\boldsymbol{B}_{(k,l)} = \sum_{r=1}^{R'} Q_{kl}^{(r)} \boldsymbol{P}^{(r)}$, with $1 \leq m, n \leq I_1$ and $1 \leq k, l \leq I_2$.

The coupled-eigenvalue algorithm is presented in Alg. 4. We can initialize the algorithm by computing $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ from the rank-1 approximation solution obtained with SeROAP, THOSVD or any other rank-1 approximation method.

The complexity per iteration of the CE algorithm is dominated by the construction of the matrices in the LHS of (2.14) and (2.15), which is of order $\mathcal{O}\{\min(I_1^2 I_2^2, I_1^2 I_3^2, I_2^2 I_3^2)\}$. Suppose I_3 is

input : $\boldsymbol{\mathcal{X}}$: rank-1 approximation of a tensor $\boldsymbol{\mathcal{T}}$
output: $\boldsymbol{\mathcal{X}}^\star$: improved rank-1 approximation
Compute $\boldsymbol{x}_0$ from $\boldsymbol{\mathcal{X}}$ as
$\boldsymbol{x}_0 \leftarrow \boldsymbol{\mathcal{X}}(:, i_2, i_3)/\|\boldsymbol{\mathcal{X}}(:, i_2, i_3)\|$ for some i_2, i_3;
$t \leftarrow 0$;
repeat
  Set $\boldsymbol{x} \leftarrow \boldsymbol{x}_t$ in eigenvalue problem (2.15) and take $\boldsymbol{y}_{t+1}$ as the eigenvector whose eigenvalue is maximum;
  Set $\boldsymbol{y} \leftarrow \boldsymbol{y}_{t+1}$ in eigenvalue problem (2.14) and take $\boldsymbol{x}_{t+1}$ as the eigenvector whose eigenvalue is maximum;
  $t \leftarrow t + 1$;
until *some stopping criterion is satisfied*;
for $i_3 = 1$ *to* I_3 **do**
  $z^\star_{i_3} \leftarrow \langle \boldsymbol{t}_{i_3}, \boldsymbol{y}_t \boxtimes \boldsymbol{x}_t \rangle$;
end
$\boldsymbol{\mathcal{X}}^\star \leftarrow \boldsymbol{x}_t \otimes \boldsymbol{y}_t \otimes \boldsymbol{z}^\star$;

Algorithm 4: CE rank-1 approximation. Above, we chose to start with $\boldsymbol{x}$, but we could equivalently have started with $\boldsymbol{y}$.

the largest dimension. If $I_3 \gg I_1 I_2$, then we can take advantage of the CE algorithm in terms of complexity in comparison with the ALS algorithm. Indeed, the complexity per iteration of ALS for rank-1 approximation is of order $\mathcal{O}(3I_1I_2I_3)$, which is higher than that of the CE algorithm in this case. For instance, let $I_1 = 5, I_2 = 10$ and $I_3 = 50$. While ALS performs 7500 operations/iteration, the computational complexity of CE is of order 2500. Notice, however, that a properly comparison makes sense if the same initialization is employed in both algorithms.

The following proposition shows that the above algorithm improves (in worst case the solution remains the same) any rank-1 approximation algorithm.

Proposition 2.4.1 *Let $\boldsymbol{\mathcal{X}}$ be a rank-1 approximation of a three-way tensor $\boldsymbol{\mathcal{T}}$ delivered by some method. If $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{X}}^\star$ are the input and output of CE algorithm, then the inequality $\|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^\star\| \leq \|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}\|$ holds.*

Proof: Plugging the expression of $\boldsymbol{A}_{(m,n)}$ into equation (2.14), we obtain, after simplifications, that $\lambda = \Gamma(\boldsymbol{x}, \boldsymbol{y})$, which is the objective function of Problem (2.13). The same result is obtained when the matrix $\boldsymbol{B}_{(k,l)}$ is plugged into equation (2.15). Now $\forall t \geq 1$, let $\lambda_t^{(\boldsymbol{x})}$ and $\lambda_t^{(\boldsymbol{y})}$ be the maximal eigenvalues whose eigenvectors are $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, respectively.

The eigenpair $(\lambda_{t+1}^{(\boldsymbol{y})}, \boldsymbol{y}_{t+1})$ obtained by solving equation (2.15) with $\boldsymbol{x} = \boldsymbol{x}_t$, is solution of the maximization problem

$$\lambda_{t+1}^{(\boldsymbol{y})} = \max_{\|\boldsymbol{y}\|_2=1} \Gamma(\boldsymbol{x_t}, \boldsymbol{y}).$$

Also, the eigenpair $(\lambda_{t+1}^{(\boldsymbol{x})}, \boldsymbol{x}_{t+1})$ obtained by setting $\boldsymbol{y} = \boldsymbol{y}_{t+1}$ in equation (2.14), is solution of the problem

$$\lambda_{t+1}^{(\boldsymbol{x})} = \max_{\|\boldsymbol{x}\|_2=1} \Gamma(\boldsymbol{x}, \boldsymbol{y_{t+1}}).$$

Since $\Gamma(\boldsymbol{x_{t+1}}, \boldsymbol{y_{t+1}}) = \max_{\|\boldsymbol{x}\|_2=1} \Gamma(\boldsymbol{x}, \boldsymbol{y_{t+1}})$, it follows in particular that

$$\Gamma(\boldsymbol{x_t}, \boldsymbol{y_{t+1}}) \leq \Gamma(\boldsymbol{x_{t+1}}, \boldsymbol{y_{t+1}}),$$

which implies that $\lambda_{t+1}^{(\boldsymbol{y})} \leq \lambda_{t+1}^{(\boldsymbol{x})}$.

Similarly, plugging $\boldsymbol{x}_{t+1}$ into equation (2.15), we can conclude that $\lambda_{t+1}^{(\boldsymbol{x})} \leq \lambda_{t+2}^{(\boldsymbol{y})}$ for the reason that

$$\Gamma(\boldsymbol{x_{t+1}}, \boldsymbol{y_{t+1}}) \leq \Gamma(\boldsymbol{x_{t+1}}, \boldsymbol{y_{t+2}}).$$

Hence, the sequence

$$\{\Gamma_t\}_{t\in\mathbb{N}} = \{\ldots, \lambda_t^{(\boldsymbol{y})}, \lambda_{t+1}^{(\boldsymbol{x})}, \lambda_{t+1}^{(\boldsymbol{y})}, \lambda_{t+2}^{(\boldsymbol{x})}, \ldots\}$$

is monotonically non-decreasing. The same conclusion would be achieved if we begin by plugging $\boldsymbol{x}_t$ into equation (2.15).

Now, let $\boldsymbol{\mathcal{X}} = \boldsymbol{x}_0 \otimes \boldsymbol{y}_0 \otimes \boldsymbol{z}_0$ be a rank-1 approximation obtained with any other method. Assume $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ are unit vectors, and define $\lambda_0 = \Gamma(\boldsymbol{x_0}, \boldsymbol{y_0})$. By setting $\boldsymbol{x} = \boldsymbol{x}_0$ in equation (2.15) in the first iteration (a similar operation would be possible for $\boldsymbol{y}_0$ in equation (2.14)), we clearly have $\lambda_0 \leq \lambda_1^{(\boldsymbol{y})} \leq \lambda_{t_{max}}^{(\boldsymbol{y})}$, where t_{max} is the iteration in which the stopping criterion is satisfied.

Since the optimization problems (2.11) and (2.13) are equivalent, $z_{i_3}^\star, 1 \leq i_3 \leq I_3$, can be obtained by performing the scalar product between vectors $\boldsymbol{t}_{i_3}$ and $\boldsymbol{y}_{t_{max}} \boxtimes \boldsymbol{x}_{t_{max}}$ (which is equivalent to contracting tensor $\boldsymbol{\mathcal{T}}$ on $\boldsymbol{x}_{t_{max}}$ and $\boldsymbol{y}_{t_{max}}$). Hence, the tensor

$$\boldsymbol{\mathcal{X}}^\star = \boldsymbol{x}_{t_{max}} \otimes \boldsymbol{y}_{t_{max}} \otimes \boldsymbol{z}^\star$$

is a better rank-1 approximation of $\boldsymbol{\mathcal{T}}$ than $\boldsymbol{\mathcal{X}}$, implying $\|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}^\star\| \leq \|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{X}}\|$. □

Corollary 2.4.2 *For any input $\boldsymbol{\mathcal{X}}$ in CE algorithm, the objective of (2.13) always converges to a limit value.*

Proof: In the proof of Proposition 2.4.1, we have shown that $\{\Gamma_t\}$ is monotonically non-decreasing for any input $\boldsymbol{\mathcal{X}}$. Let $p^\star$ be the maximum of the objective (2.13). Since the best rank-1 approximation problem always has a solution, then $p^\star < \infty$. But $\max_{\boldsymbol{x}} \Gamma(\boldsymbol{x}, \boldsymbol{y_{t+1}}) \leq \max_{\boldsymbol{x},\boldsymbol{y}} \Gamma(\boldsymbol{x}, \boldsymbol{y})$, which implies that $\{\Gamma_t\}_{t\in\mathbb{N}}$ is bounded above by $p^\star$. Since $\{\Gamma_t\}$ is a real non decreasing sequence bounded above, it converges to a limit $\Gamma^\star$, $\Gamma^\star \leq p^\star$. □

2.4.2 Performance of iterative methods

The next section deals with the performance of CE algorithm in some simulated scenarios. We compare for a sample of complex tensors our CE method with the standard ALS algorithm using the estimated approximation error. We also determine the average iteration in which both algorithms satisfy a stopping criterion.

Performance CE and ALS methods in general scenarios

In order to compare CE and ALS algorithms, we assume that both algorithms are initialized with the same factors. We assume two types of initializations: random and that delivered by SeROAP method running with the permutation vector $\boldsymbol{p} = [1\ 2\ 3]$. The performance of the algorithms are evaluated in four scenarios: $3 \times 4 \times 5$, $3 \times 4 \times 20$, $3 \times 20 \times 20$ and $20 \times 20 \times 20$ complex tensors. As performed in Section 2.3.2, the choice of scenarios intends to account for the variability of tensor dimensions. A sample of 300 tensors with real and imaginary parts distributed according to a uniform measure in $[-1, 1]$ was generated. The stopping criterion of the algorithms is the maximum number of iterations ($it_{max} = 200$) or $|E_{it} - E_{it-1}| \leq 10^{-6}$, where E_{it} is the approximation error at it-th iteration.

Let $\Delta\phi^{[n]} = \|\boldsymbol{\mathcal{T}}^{[n]} - \boldsymbol{\mathcal{X}}^{[n]}_{ALS}\| - \|\boldsymbol{\mathcal{T}}^{[n]} - \boldsymbol{\mathcal{X}}^{[n]}_{CE}\|$ be the difference between the approximation errors of tensor n of the sample obtained with ALS and CE. Notice that if $\Delta\phi^{[n]} \geq 0$, then CE method performs better than ALS for the tensor $\boldsymbol{\mathcal{T}}^{[n]}$.

Tables below present the percentage of tensors in which $\Delta\phi^{[n]} \geq 0$ and the mean iteration in which the stopping criterion is satisfied for two types of initializations. We have rounded the mean iteration to the nearest integer.

Scenario	$\%\ \Delta\phi^{[n]} \geq 0$	mean iteration (ALS)	mean iteration (CE)
$3 \times 4 \times 5$	91.00 %	27	13
$3 \times 4 \times 20$	87.00 %	44	14
$3 \times 20 \times 20$	86.67 %	71	27
$20 \times 20 \times 20$	62.67 %	102	52

Table 2.2: Performance of CE method compared to ALS with a random initialization.

Scenario	$\%\ \Delta\phi^{[n]} \geq 0$	mean iteration (ALS)	mean iteration (CE)
$3 \times 4 \times 5$	97.33 %	23	10
$3 \times 4 \times 20$	96.00 %	36	11
$3 \times 20 \times 20$	91.33 %	57	19
$20 \times 20 \times 20$	78.00 %	86	42

Table 2.3: Performance of CE method compared to ALS initialized with SeROAP.

Notice that in the three first scenarios with the SeROAP initialization, CE method performs better than ALS for most of tested tensors. In last scenario, the performance of CE is still better than that of ALS but for less tensors.

For the random initialization, CE carries on with the better performance, but the percentage of $\Delta\phi^{[n]}$ decrease in all scenarios.

As expected the mean iteration is greater for the scenario with random initialization. Additionally, CE algorithm converges faster than ALS on average in all cases.

2.5 Chapter Summary and Directions

We conclude this chapter by presenting the main points observed in the results and also we given some directions for future work.

- *Lasserre's approach for computing rank-1 approximations.* Despite the fact that the moment approach provides a certificate of optimality to verify whether the estimated solution delivered by the relaxed problem (A.5) is optimum (which is true in general), the method is inefficient even for medium-sized tensors. This is the reason why only $2 \times 2 \times 2$ scenarios were simulated. As a metter of fact, we confirmed that computation of the best rank-1 approximation for a $3 \times 3 \times 3$ tensor already requires a lot of time (about 30 minutes). In order to mitigate this drawback in future work, we incite the use of the algorithm proposed in [Bucero 2014], which improves Lasserre's method. However, we can not expect a meaningful improvement in terms of computational time for larger tensors, since the exponential increasing of variables due to relaxations is still an inconvenience. Thus, since the monomials of the polynomial describing the objective of the best rank-1 approximation problem are sparse, we expect that optimization can still be done in order to reduce the computational time of the rank-1 estimation.

- *SeROAP method*. Our proposed method was proved useful for computing rank-1 approximations of three-way tensors. Indeed, we gave a mathematical proof for three-way tensors in which the approximation error is at most equal to those computed with standard methods, such as the truncated HOSVD (THOSVD) and the more recently method called ST-HOSVD. For fourth order tensors, we confirmed by simulations that SeROAP still brings some advantages over THOSVD. However, for higher order tensors, we saw that its performance is hugely degraded. Additionally, SeROAP allowed us to compute a rank-1 approximation closer than the best one in $2 \times 2 \times 2$ scenarios, when compared to the other methods. Unfortunately, we could not see how far the approximation errors delivered by the mentioned finite methods are from the best rank-1 approximation for tensors with larger dimensions, due to limitations of Lasserre's method for computing solutions in a reasonable time. The development of more efficient algorithms allowing us to compute the best rank-1 approximation faster is still a challenge.

- *CE algorithm*. Although this method has better computational complexity than ALS only in specific conditions (one dimension must be large enough than the other two dimensions), the approximation error and the average number of iterations to convergence obtained with CE is better than those computed with ALS. These results showed that CE is a very competitive iterative method to compute rank-1 approximations for three-way tensors. In the future, we aim to analyze some convergence aspects of CE algorithm. For instance, we still do not have any theoretical result showing that CE algorithm converges to a unique stationary point for any initialization (we have only numerical proofs).

CHAPTER 3

Iterative Deflation

This chapter presents a detailed study of iterative deflation algorithms based on finite rank-1 approximations with the goal of computing low rank tensor approximations or the exact CP decomposition.

First, we introduce the general concept of deflation and present some examples of its usage in some mathematical domains. Specifically, we present the iterative deflation concept on low rank tensor approximations. Next, we perform a detailed description of the so-called DCPD algorithm, whose concept is based on the computation of successive rank-1 tensor approximations and deflation procedures. Contrary to others deflation algorithms, called in the literature as hierarchical algorithms, we use within DCPD the finite rank-1 approximation algorithms described in Chapter 2.

In the sequel, we perform a theoretical study on iterative deflation in order to analyze the convergence of the DCPD algorithm assuming a best rank-1 approximation to update rank-1 components. Some important lemmas, propositions and a conjecture are presented to ensure the convergence of DCPD algorithm under some conditions for the exact CP decomposition.

In the last part of this chapter, we draw some numerical experiments. In a first stage, we evaluate the percentage of successful decompositions of the DCPD algorithm for the three rank-1 approximations algorithms: SeROAP, THOSVD and ST-HOSVD. We compare the performance with the standard ALS and the iterative deflation algorithm HALS. Next, for the same algorithms, we analyze the behavior of residuals and convergence under noisy scenarios, and the average approximation error under rank and order variation. The chapter is closed with some conclusion and perspectives of future works.

Main contributions of this chapter:

- *DCPD algorithm.* We propose an iterative deflation algorithm to compute low rank tensor approximations. The idea of iterative deflation algorithms, originally called hierarchical algorithms, was already introduced in the literature, but here we replace the original alternating update of factors of the known HALS algorithm by finite rank-1 tensor approximation algorithms, with the goal of reducing residuals in a few iterations. Our algorithm is called DCPD, stands for Deflation Canonical Polyadic Decomposition.

- *Theoretical study on the convergence of DCPD algorithm.* We show that the norm of residuals is monotonically reduced within the iterative deflation process. We also prove that the DCPD algorithm recovers the exact CP decomposition of a given tensor when residuals do not fall within a cone with an arbitrary small volume. In a second stage, we prove that the iterative deflation method can reduce the norm of the initial residual by a factor smaller than $(\sin(\beta))^{L-1}$ (β being the angle of a suitable cone where the residuals can fall in) after L iterations with

high probability, when tensors are distributed according to an absolutely continuous probability measure, and the probability function of residuals is continuous on some suitable angular interval. We also present a conjecture stating the existence of probability measures ensuring the convergence of the DCPD algorithm to an exact decomposition with high probability.

- *Performance of DCPD algorithm.* Our conjecture is reinforced for DCPD algorithm by means of the computation of CP decompositions of tensors with the entries distributed according to a uniform distribution. In general, we also verify by simulations that residuals decrease faster with DCPD + rank-1 approximation algorithm, when compared to the HALS algorithm. We also present a framework of experiments for noisy and noiseless scenarios with rank and order variations.

Contents

3.1 Deflation on CP decomposition

Generally speaking, deflation is a computational technique applied to a wide range of mathematical problems whose purpose is to remove some mathematical features and hence to deal with less complex problems. As a simple example, consider that some algorithm (Newton, gradient, etc) is employed to compute a root α of a univariate polynomial $p(x)$. This root can be eliminated (*defleted*) from the polynomial $p(x)$ by performing

$$q(x) = \frac{p(x)}{x - \alpha},$$

where $\deg(q(x)) = \deg(p(x)) - 1$. Clearly, the roots of $q(x)$ are also roots of the polynomial $p(x)$, but now one root had been removed the division by $x - \alpha$. The same algorithm combined with this *deflation* procedure can be sucessively applied to obtain all other roots of $p(x)$.

Deflation also takes place in the improvement of conjugate gradient methods, in which a subspace is hidden (*deflated*) from the algoritm itself in order to improve its convergence speed. See [Nicolaides 1987] for a longer discussion.

A last example is the computation of rank-r approximations of matrices. It is known that the rank-r approximation of an $m \times n$ matrix $\boldsymbol{A}$ can be obtained from its r-dominant singular triplets $(\sigma_i, \boldsymbol{u}_i, \boldsymbol{v}_i),\ 1 \leq i \leq r$, with $r \leq \min(m, n)$. As a metter of fact,

$$\sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathsf{H}} = \underset{\boldsymbol{X}, \mathrm{rank}\{\boldsymbol{X}\}=r}{\arg\min} \|\boldsymbol{A} - \boldsymbol{X}\|.$$

For large-sized matrices, the computation of r-singular triplets all-at-once can be an issue, so that it is suitable to compute part of them. For instance, we can compute one-by-one the singular triplets with deflations by subtraction. Thus, the rank-1 approximation of $\boldsymbol{A}$, namely $\sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathsf{H}}$, can be computed with some algorithm (e.g. Lanczo's [Comon 1990]) and *deflated* from $\boldsymbol{A}$, resulting in the matrix $\boldsymbol{A} - \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathsf{H}}$, on which the same algorithm can be used in order to recover the second dominant singular triplet $(\sigma_2, \boldsymbol{u}_2, \boldsymbol{v}_2)$. Hence, by repeating r-times both the deflation and the rank-1 approximation, we recover all r-singular triplets, thereby constructing the rank-r approximation of $\boldsymbol{A}$.

The question that arises is whether a deflation property similar to that applied to matrices can be extended to tensors. In other words, we wonder if the best rank-1 approximation of a tensor $\boldsymbol{\mathcal{T}}$ is a rank-1 component of its decomposition. The answer is no in general, and it is shown in [Stegeman 2010]. Therein, the authors show that subtracting a best rank-1 approximation from a tensor can even increase the rank of the resulting tensor. However, if the tensor $\boldsymbol{\mathcal{T}}$ is symmetric and orthogonally decomposable, then the deflation procedure can be applied to [Zhang 2001].

Herein, we introduce the concept of *iterative deflation*, which is defined as the computational iterative technique whose purpose is the extraction of some mathematical features of some problem at any given iteration to be used at the next iteration. Notice that contrary to the preceding definition of deflation, we do not remove the mathematical features, but they are entries of a next problem. This techique can be used to compute the low rank approximation of tensors, as we will present in Section 3.1.1.

In order to understand the iterative deflation procedure, let $\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \boldsymbol{\mathcal{X}}_r + \boldsymbol{\mathcal{E}}$ be a decomposition of a tensor $\boldsymbol{\mathcal{T}}$, where $\boldsymbol{\mathcal{X}}_r,\ 1 \leq r \leq R$, are rank-1 tensors, and $\boldsymbol{\mathcal{E}}$ is a residual tensor. At iteration

1, we compute the best rank-1 approximation of $\boldsymbol{\mathcal{X}}_1 + \boldsymbol{\mathcal{E}}$, namely $\boldsymbol{\mathcal{X}}'_1$, and extract the residual $\boldsymbol{\mathcal{E}}' = \boldsymbol{\mathcal{X}}_1 + \boldsymbol{\mathcal{E}} - \boldsymbol{\mathcal{X}}'_1$. At iteration 2, we add the residual $\boldsymbol{\mathcal{E}}'$ to tensor $\boldsymbol{\mathcal{X}}_2$, and compute again the rank-1 approximation but now of the tensor $\boldsymbol{\mathcal{X}}_2 + \boldsymbol{\mathcal{E}}'$, and extract another residual to be used at iteration 3, and so on. This simple procedure is actually the idea behind of DCDP algorithm, which is explained in details later.

The DCPD algorithm is similar to the nonnegative tensor factorization algorithm, called *simple* HALS NTF, described in [Cichocki 2009c]. Therein, the algorithm computes the nonnegative factors $\boldsymbol{A}_r,\ 1 \leq r \leq R$ of a nonnegative tensor $\boldsymbol{\mathcal{Y}}$, where R is the rank of the approximating tensor. They use an alternating least square procedure for updating each column of each $\boldsymbol{A}_r$, with the negative entries projected to $\mathbb{R}_+$. This method can be extend to ordinary tensors by not performing the projection, in which case we call it simply HALS. These authors prefer to use the terminology *hierarchical* instead of iterative deflation. We do not agree with the former terminology since there is no actually hierarchical relation in their methods. Contrary to HALS, the DCPD algorithm does not use an alternating least square procedure to update rank-1 tensors at every iteration. For this task, we use rank-1 approximation algorithms, such as SeROAP, THOSVD and ST-HOSVD. The goal is to reduce quickly the norm of residuals according as the iteration increases, which not happens with HALS algorithm (see Section 3.3.2). Other tensor deflation algorithms can be found in [Cichocki 2009b, Phan 2015a, Phan 2015b].

In the following, we present a detailed description of DCPD algorithm in the context of low rank tensor approximation.

3.1.1 Description of the DCPD algorithm

The DCPD is an iterative deflation algorithm [da Silva 2015] that computes low rank tensor approximations for real or complex tensors. Contrary to the so-called Hierarchical ALS algorithm proposed in [Cichocki 2009c], rank-1 components are updated by using rank-1 approximation algorithms. The DCPD algorithm goes along the lines depicted in Alg.5.

In the first *for* loop, the rank-1 tensors $\boldsymbol{\mathcal{X}}[1,1], \ldots, \boldsymbol{\mathcal{X}}[R,1]$ are computed by successive rank-1 approximations and subtractions. Since the rank of the tensor does not decrease with subtractions in general [Stegeman 2010], a residual $\boldsymbol{\mathcal{E}}[R,1]$ is then produced. In the iterative process (*repeat* loop), a new rank-1 component, namely $\boldsymbol{\mathcal{X}}[1,2]$, is computed from the sum of the previous residual and $\boldsymbol{\mathcal{X}}[1,1]$. The tensor $\boldsymbol{\mathcal{Y}}[1,2]$ is updated within the *if-else* condition, and a new residual $\boldsymbol{\mathcal{E}}[1,2]$ is produced with the subtraction $\boldsymbol{\mathcal{Y}}[1,2] - \boldsymbol{\mathcal{X}}[1,2]$. By applying the same procedure to the other components, we update all R rank-1 tensors, so that another residual $\boldsymbol{\mathcal{E}}[R,2]$ is generated at the end of the second *for* loop. The second loop continues to execute until some stopping criterion is satisfied, and hence R rank-1 components are computed for $\boldsymbol{\mathcal{T}}$. If $\mathrm{rank}\{\boldsymbol{\mathcal{T}}\} \leq R$, then an exact decomposition can be recovered.

3.2 Study on iterative deflation and best rank-1 tensor approximations

In [da Silva 2015], we proved that the normalized residual $(\|\boldsymbol{\mathcal{E}}[R,l]\|)_{l \in \mathbb{N}_{>0}}$ is a monotonically decreasing sequence when the best rank-1 approximation is assumed within DCPD. In this section, a thorough theoretical study is presented. Based on a geometric approach, we sketch an analysis of the

input : $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: input data,
R: rank parameter.
ϕ: an algorithm computing a rank-1 approximation
output: $\boldsymbol{\mathcal{X}}_r \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$, for $r = 1, \ldots, R$: rank-1 components
$\boldsymbol{\mathcal{Y}}[1,1] \leftarrow \boldsymbol{\mathcal{T}}$;
for $r = 1$ *to* R **do**
  $\boldsymbol{\mathcal{X}}[r,1] \leftarrow \phi(\boldsymbol{\mathcal{Y}}[r,1])$;
  if $r < R$ **then**
    $\boldsymbol{\mathcal{Y}}[r+1,1] \leftarrow \boldsymbol{\mathcal{Y}}[r,1] - \boldsymbol{\mathcal{X}}[r,1]$;
  else
    $\boldsymbol{\mathcal{E}}[R,1] \leftarrow \boldsymbol{\mathcal{Y}}[R,1] - \boldsymbol{\mathcal{X}}[R,1]$;
  end
end
$l \leftarrow 2$;
repeat
  for $r = 1$ *to* R **do**
    if $r > 1$ **then**
      $\boldsymbol{\mathcal{Y}}[r,l] \leftarrow \boldsymbol{\mathcal{X}}[r,l-1] + \boldsymbol{\mathcal{E}}[r-1,l]$;
    else
      $\boldsymbol{\mathcal{Y}}[1,l] \leftarrow \boldsymbol{\mathcal{X}}[1,l-1] + \boldsymbol{\mathcal{E}}[R,l-1]$;
    end
    $\boldsymbol{\mathcal{X}}[r,l] \leftarrow \phi(\boldsymbol{\mathcal{Y}}[r,l])$;
    $\boldsymbol{\mathcal{E}}[r,l] \leftarrow \boldsymbol{\mathcal{Y}}[r,l] - \boldsymbol{\mathcal{X}}[r,l]$;
  end
  $l \leftarrow l + 1$;
until *some stopping criterion is satisfied*;
foreach $r \in [1, \ldots, R]$ **do**
  $\boldsymbol{\mathcal{X}}_r \leftarrow \boldsymbol{\mathcal{X}}[r,l]$;
end

Algorithm 5: DCPD algorithm

convergence of the DCDP algorithm, including a conjecture that it converges to an exact decomposition with high probability when tensors within $\mathcal{T}^{(R)} = \{\boldsymbol{\mathcal{T}} \in \mathcal{T} : \operatorname{rank} \boldsymbol{\mathcal{T}} \leq R\}$ are distributed according to some absolutely continuous probability measure.

First, let us take a closer look at the 2D geometric interpretation of the DCPD algorithm. Figure 3.1 depicts the $[r,l]$-iteration for $r > 1$, so that $\gamma[r,l]$ is the angle between the tensors $\boldsymbol{\mathcal{E}}[r-1,l]$ and $\boldsymbol{\mathcal{X}}[1,l-1]$. For $r = 1$, the residual $\boldsymbol{\mathcal{E}}[r-1,l]$ can be just replaced with $\boldsymbol{\mathcal{E}}[R,l-1]$ in the figure, and $\gamma[1,l]$ is then defined from $\boldsymbol{\mathcal{E}}[R,l-1]$ and $\boldsymbol{\mathcal{X}}[1,l-1]$. The meaning of angle β and the gray area in the figure will be explained later.

Before stating some theoretical results on the DCPD algorithm, we present a fundamental lemma related to the error in rank-1 approximations of tensors of the form $\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{E}}$, where $\boldsymbol{\mathcal{X}}$ is a rank-1 tensor and $\boldsymbol{\mathcal{E}}$ any other tensor, both with entries in some field $\mathbb{K}$.

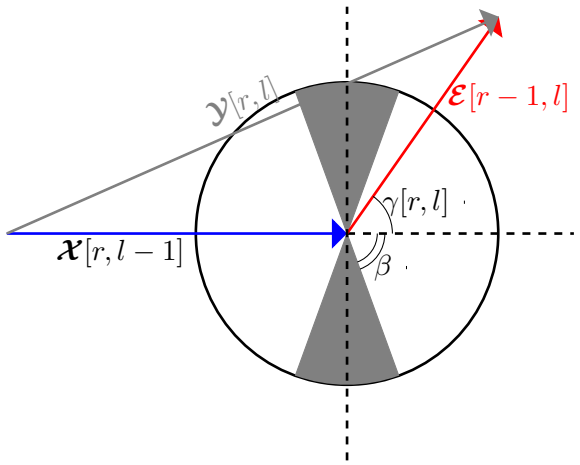


Figure 3.1: Visualization of the residual in an n-sphere for some iteration l of DCPD algorithm.

Lemma 3.2.1 *Let $\mathcal{X}$ be a rank-1 tensor and ϕ the best rank-1 approximation operator. For any tensor $\mathcal{E}$,*

$$\|\mathcal{X}+\mathcal{E}-\phi(\mathcal{X}+\mathcal{E})\| \leq \sin(\gamma)\|\mathcal{E}\|,$$

where γ denotes the angle between $\mathcal{E}$ and $\mathcal{X}$.

Proof: Let $\mathcal{P}_{\mathcal{X}}(\mathcal{X}+\mathcal{E})$ be the orthogonal projection of $\mathcal{X}+\mathcal{E}$ onto $\operatorname{span}(\mathcal{X})$. Because $\phi(\mathcal{X}+\mathcal{E})$ is a best rank-1 approximation of $\mathcal{X}+\mathcal{E}$, $\mathcal{P}_{\mathcal{X}}(\mathcal{X}+\mathcal{E})$ cannot be a strictly better rank-1 approximation than $\phi(\mathcal{X}+\mathcal{E})$. Thus,

$$\|\mathcal{X}+\mathcal{E}-\phi(\mathcal{X}+\mathcal{E})\| \leq \|\mathcal{X}+\mathcal{E}-\mathcal{P}_{\mathcal{X}}(\mathcal{X}+\mathcal{E})\|.$$

On the other hand, $\mathcal{X}+\mathcal{E}-\mathcal{P}_{\mathcal{X}}(\mathcal{X}+\mathcal{E}) \perp \mathcal{X}$. Hence, we have $\|\mathcal{X}+\mathcal{E}-\mathcal{P}_{\mathcal{X}}(\mathcal{X}+\mathcal{E})\| = \sin(\gamma)\|\mathcal{E}\|$ by using basic trigonometry. This concludes the proof. □

The following results for the DCPD algorithm stems from the previous lemma.

Corollary 3.2.2 *In DCPD algorithm, the inequality $\|\mathcal{E}[r,l]\| \leq \sin(\gamma[r,l])\|\mathcal{E}[r-1,l]\|$ holds for any $1 < r \leq R$.*

Proof: By replacing $\mathcal{X}, \mathcal{E}$ and γ in Lemma 3.2.1 with $\mathcal{X}[r,l-1], \mathcal{E}[r-1,l]$ and $\gamma[r,l]$ respectively, the result follows directly. □

Corollary 3.2.3 *For any $l > 1$ and $c_l = \prod_{r=1}^{R} \sin(\gamma[r,l])$, the inequality $\|\mathcal{E}[R,l]\| \leq c_l\|\mathcal{E}[R,l-1]\|$ holds.*

Proof: By applying $R-1$ times the result of Corollary 3.2.2, we have

$$\|\mathcal{E}[R,l]\| \leq \left(\sin(\gamma[R,l]) \cdots \sin(\gamma[2,l])\right) \|\mathcal{E}[1,l]\|.$$

From Lemma 3.2.1, we know that

$$\|\boldsymbol{\mathcal{E}}[1,l]\| \leq \sin(\gamma[1,l])\|\boldsymbol{\mathcal{E}}[R,l-1]\|.$$

Thus, it follows that $\|\boldsymbol{\mathcal{E}}[R,l]\| \leq c_l\|\boldsymbol{\mathcal{E}}[R,l-1]\|$. □

Notice that the same result brought by Proposition (4.4) in [da Silva 2015] can be deduced from Corollary 3.2.3 since $0 \leq c_l \leq 1$ for every iteration l, which implies the monotonic decrease of the sequence $\{\|\boldsymbol{\mathcal{E}}[R,l]\|\}_{l\in\mathbb{N}_{>0}}$.

Corollary 3.2.4 *If* $\|\boldsymbol{\mathcal{E}}[R,l]\| = \|\boldsymbol{\mathcal{E}}[R,l-1]\|$, *then* $c_l = 1$.

Proof: Because $c_l \leq 1$ and $\|\boldsymbol{\mathcal{E}}[R,l]\| = \|\boldsymbol{\mathcal{E}}[R,l-1]\|$, one concludes directly from Corollary 3.2.3 that $c_l = 1$. □

From Corollary 3.2.4, we note that DCPD might not improve the estimation of the rank-1 components anymore for $l \geq l_0 > 1$. And this may occur not only in the presence of noise. As a metter of fact, even for an almost orthogonal case $c_l \approx 1$, $\|\boldsymbol{\mathcal{E}}[R,l]\|$ may tend to a stationary non-zero value as l increases. However, the DCPD algorithm converges to an exact decomposition if $c_l \leq C$, for all $l > 1$, and some constant $C < 1$. This will be subsequently detailed by means of a geometric approach.

Figure 3.1 can also be seen as the representation of an n-sphere of dimension $n = I_1 I_2 \cdots I_N - 1$ in $\mathbb{K}^{n+1}$ space. β is half the white cone angle defined in $[0, \pi/2]$. The direction of the rank-1 tensor $\boldsymbol{\mathcal{X}}[r,l-1]$ defines the axis of the white cone and varies with r or l. Thus, the cone axis changes at every iteration $[r,l]$ but the opening of the white cone remains the same. Under a condition on β, we can state an important proposition ensuring the convergence of the DCPD algorithm.

Proposition 3.2.5 *Let* $\boldsymbol{\mathcal{T}}$ *be a tensor such that* $\operatorname{rank}\boldsymbol{\mathcal{T}} \leq R$. *An exact decomposition is recovered by the DCPD algorithm if and only if there exists for every* (r,l) *a half cone of angle* β *(in white in Fig. 3.1),* $0 \leq \beta < \pi/2$, *such that*

$$\beta \geq \max_{l>1} \min_{1\leq r\leq R} \gamma[r,l].$$

Proof: ($\Leftarrow$) For any iteration $l > 1$, take $\gamma[r_0,l] = \min_{1\leq r\leq R}\gamma[r,l]$. Notice that $c_l \leq \sin(\gamma[r_0,l])$ from Corollary 3.2.3. By hypothesis, $\sin(\gamma[r_0,l]) \leq \sin(\beta)$, which implies that $\|\boldsymbol{\mathcal{E}}[R,l]\| \leq c_l\|\boldsymbol{\mathcal{E}}[R,l-1]\| \leq \sin(\beta)\|\boldsymbol{\mathcal{E}}[R,l-1]\|$. Because β is an upper bound for $\gamma[r,l]$, $l > 1$, we have $\sin(\beta)\|\boldsymbol{\mathcal{E}}[R,1]\| \geq \|\boldsymbol{\mathcal{E}}[R,2]\| \implies (\sin(\beta))^{l-1}\|\boldsymbol{\mathcal{E}}[R,1]\| \geq \|\boldsymbol{\mathcal{E}}[R,l]\|$. Hence, when $l \to \infty$, $\|\boldsymbol{\mathcal{E}}[R,l]\| \to 0$.

($\Rightarrow$) Let l_0 be some iteration such that $l_0 > 1$. Without loss of generality, assume $\|\boldsymbol{\mathcal{E}}[R,l]\| = 0$ for $l \geq l_0$ (l_0 can be arbitrarily large). Then $(\|\boldsymbol{\mathcal{E}}[R,l]\|)_{l\in\mathbb{N}_{>0}}$ is a strictly monotonically decreasing sequence for $1 < l < l_0$, otherwise the algorithm would converge to a nonzero constant for some iteration smaller than l_0. Hence, for every (r,l) we can choose β, $0 \leq \beta < \pi/2$ such that $\beta \geq \max_{l>1} \min_{1\leq r\leq R} \gamma[r,l]$, and the proof is complete. □

As a conclusion, if for a given iteration l, all tensors $\boldsymbol{\mathcal{E}}[r,l]$, $1 \leq r \leq R$, fall within the gray volume depicted in Fig. 3.1 (the complementary of the white cone), then the sequence $\boldsymbol{\mathcal{E}}[r,l]$ does not tend to

zero. Even if this gray volume can be made arbitrarily small, it is not of zero measure. In general, the best we can do is to study the convergence of the DCPD algorithm to an exact decomposition under some probabilistic conditions.

Lemma 3.2.6 *If tensors $\boldsymbol{\mathcal{T}}$ are distributed within $\mathcal{T}^{(R)}$ according to an absolutely continuous probability measure, then $\|\boldsymbol{\mathcal{E}}[r,l]\|$ are absolutely continuous random variables.*

Proof: Let $\boldsymbol{D} = [I_1 \cdots I_N]$ be a specific size of n-order tensors, and let $\mathcal{T}_{\boldsymbol{D}}^{(R)} = \{\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times \cdots \times I_N} : \boldsymbol{\mathcal{T}} \subset \mathcal{T}^{(R)}\}$. Because $\mathcal{T}^{(R)} \supset \mathcal{T}_{\boldsymbol{D}}^{(R)}$, any tensor $\boldsymbol{\mathcal{T}}$ within $\mathcal{T}_{\boldsymbol{D}}^{(R)}$ is also distributed according to an absolutely continuous probability measure. Via the DCDP algorithm, each rank-1 component obtained in successive deflations is also in $\mathcal{T}_{\boldsymbol{D}}^{(R)}$. Hence, since the sum (subtraction) of continuous random variables does not affect the continuity, the residuals $\boldsymbol{\mathcal{E}}[r,l]$ are also absolutely continuous random variables. Since the norm is a C^0 function in finite dimension, $\|\boldsymbol{\mathcal{E}}[r,l]\|$ is also absolutely continuous. □

For the next developments, let $Z_l = \|\boldsymbol{\mathcal{E}}[R,l]\|$ and define the following probability for some iteration $L > 1$:

$$F_L[\beta] = P\left(Z_L \leq \sin(\beta) Z_{L-1} \leq \ldots \leq (\sin(\beta))^{L-1} Z_1\right). \tag{3.1}$$

$F_L[\beta]$ can be viewed as the probability that residuals fall within at least one of the R white cones in every iteration $l \leq L$.

The following proposition ensures a reduction of Z_1 by a factor smaller than $(\sin(\beta))^{L-1}$ after L iterations with high probability, if a condition on the continuity of $F_L[\beta]$ is assumed.

Proposition 3.2.7 *Let L be fixed. If $\exists \beta_0 : \beta_0 \in [0, \pi/2)$ such that $F_L[\beta]$ is continuous on $[\beta_0, \pi/2]$, then $\forall \varepsilon : \varepsilon \in (0,1], \exists \beta \in [\beta_0, \pi/2)$ such that $F_L[\beta] > 1 - \varepsilon$.*

Proof: Since $F_L[\pi/2] = 1$ and $F_L[\beta]$ is continuous on $[\beta_0, \pi/2]$, the proof follows directly from the intermediate value theorem. □

Although $Z_l, 1 \leq l \leq L$, are absolutely continuous random variables and $g_m(\beta) = \sin^m(\beta)$ are continuous functions for all $m \geq 0$, the continuity of $F_L[\beta]$ in β is not guaranteed due to the dependence of the random variables $Z_1, \ldots, Z_L$ (Z_l depends on Z_{l-1}). For example, for $L = 2$ and $Z_1 = 2Z_2$ with probability 1, it is easy to check that $F_2[\beta]$ is not continuous at $\beta = \pi/2$. Indeed, $\lim_{\beta \to \pi/2^-} F_2[\beta] = 0$ whereas $F_2[\pi/2] = 1$.

Now, let β_l be the half-angle of the white cone at every iteration l. Contrary to the first definition of β, the angle value is not fixed anymore for all iteration $[r,l]$, but only along the r-iteration. In this case, equation 3.1 becomes

$$F_L[\beta_L] = P\left(Z_L \leq \sin(\beta_L) Z_{L-1} \leq \prod_{i=L-1}^{L} (\sin(\beta_i))\, Z_{L-1} \leq \ldots \leq \prod_{i=2}^{L} (\sin(\beta_i))\, Z_1\right). \tag{3.2}$$

The following conjecture claims that there exists absolutely continuous distributions of tensors in $\mathcal{T}^{(R)}$ such that the probability $F_l[\beta_l]$ tends to 1 as $l \to \infty$, and at the same time the norm of residuals tends to 0, which is suitable for the convergence of the DCPD algorithm to an exact CP decomposition.

Conjecture 3.2.8 *There exists absolutely continuous probability measures μ for tensors $\boldsymbol{\mathcal{T}}$ within $\mathcal{T}^{(R)}$ for which the following holds:*

(i). $\forall \varepsilon : \varepsilon \in (0,1]$, $\forall l : l > 1$, $\exists \beta_l : \beta_l \in [0, \pi/2)$, *such that* $F_l[\beta_l] > 1 - \varepsilon$.

(ii). $\forall l : l > 1$, $\exists \beta_l : \beta_l \in [0, \pi/2)$, *such that* $\prod_{i=2}^{l} \sin(\beta_i)$ *is a strictly monotonically decreasing sequence converging to* 0.

Conjecture 3.2.8 shows that for some probability distributions of $\boldsymbol{\mathcal{T}}$ within $\mathcal{T}^{(R)}$, the residual norm variable $Z_l \to 0$ as $l \to \infty$ with high probability. Indeed, subsequent computer simulations support the existence of a uniform probability measure μ for the entries of tensors within $\mathcal{T}^{(R)}$, such that $F_L[\beta_L] \approx 1$ and $Z_L \approx 0$ for large values of L. This reinforces our conjecture.

3.3 Estimation of the CP decomposition

In this section, we present some numerical experiments using the iterative deflation algorithm DCPD. The rank-1 components are updated with three finite rank-1 approximation algorithms: SeROAP, ST-HOSVD and THOSVD. We also include in our simulations the standard ALS and the HALS algorithms for the following reasons: the former is known as the "workhorse" algorithm to compute low rank tensor approximations [Kolda 2009]; the latter is the original iterative deflation algorithm extended to complex tensors.

3.3.1 Complexity of the algorithms

Since DCPD is not a finite algorithm, the total complexity is unbounded. Therefore, we have chosen the number of multiplications by iteration as the complexity metric. The complexity is mainly dominated by the rank-1 approximation function ϕ, which is computed R times. Table 3.1 summarizes the number of operations per iteration of the DCPD algorithm for three finite rank-1 approximation algorithms: SeROAP, ST-HOSVD and THOSVD; and also those of the standard ALS and HALS algorithms.

In Table 3.2, we present the complexity/iteration of the algorithms for all scenarios that will be evaluated in next sections. We assume $k = 4$. Although in simulations we have defined the ordering vector $\boldsymbol{p} = [1\ 2\ \ldots\ N]$ for DCPD-STHOSVD and DCPD-SeROAP, we do to need to take it into account here, because the tensors have equal dimensions in all scenarios. To obtain the real value of the complexity per iteration, one should multiply the values on the table by R. From Table 3.2, the order of complexity for the studied scenarios can be establish as

ALS $\prec$ HALS $\prec$ DCPD-STHOSVD$\prec$DCPD-SeROAP$\prec$DCPD-THOSVD,

where $\prec$ means "less complex than". Notice that DCPD-STHOSVD and DCPD-SeROAP algorithms have approximately the same complexities.

Algorithm	complexity/iteration
ALS	$\mathcal{O}\{3R\prod_{j=1}^{N} I_j\}$
HALS	$\mathcal{O}\{(N+2)R\prod_{j=1}^{N} I_j\}$
DCPD-THOSVD	$\mathcal{O}\{(2Nk+2)R\prod_{j=1}^{N} I_j\}$
DCPD-STHOSVD	$\mathcal{O}\{2kR\sum_{n=1}^{N-1}\prod_{i=n}^{N} I_{p_i}+2R\prod_{j=1}^{N} I_j\}$
DCPD-SeROAP	$\mathcal{O}\{(2k+2)R\sum_{n=1}^{N-1}\prod_{i=n}^{N} I_{p_i}\}$

Table 3.1: Number of operations (multiplications) per iteration of tensor algorithms.

Scenario	ALS	HALS	DCPD-SeROAP	DCPD-STHOSVD	DCPD-THOSVD
$3\times3\times3$	81	135	360	342	702
$5\times5\times5$	375	625	1500	1450	3250
$7\times7\times7$	1029	1715	3920	3822	8918
$9\times9\times9$	2187	3645	8100	7938	18954
$3\times3\times3\times3$	243	486	1170	1098	2754
$3\times3\times3\times3\times3$	729	1701	3600	3366	10206
$3\times3\times3\times3\times3\times3$	2187	5832	10890	10170	36450

Table 3.2: Complexity per iteration /R of tensor algorithms.

3.3.2 Percentage of successful decompositions

Table 3.3 presents the percentage of successful decompositions of rank-3 tensors for the algorithms ALS, HALS, DCPD-THOSVD, DCPD-STHOSVD, and DCPD-SeROAP. The ALS and HALS algorithms are randomly initialized (In [Cichocki 2009c], the authors also initialize HALS with ALS, which case will be not considered here for fairness, otherwise DCPD could also be initialized in the same way). Noise is not considered in this case so that the performance is evaluated for the computation of an exact decomposition of 500 complex tensors whose real and imaginary parts are distributed uniformly in $[-1, 1]$. We consider that a decomposition is succeeded if the residual $\|\boldsymbol{\mathcal{E}}\| \leq 10^{-6}$, which is one of the stopping criteria for the algorithms. The algorithms also stop running when a maximum of 1000 iterations is attained.

The main conclusions from the table are the following:

- The iterative deflation algorithm DCPD presents similar performance for all finite rank-1 approximations, mainly for scenarios with larger dimensions;
- HALS presents the worst performance, mainly for the scenarios with smaller dimensions;

Algorithm	$3 \times 3 \times 3$	$5 \times 5 \times 5$	$7 \times 7 \times 7$	$9 \times 9 \times 9$
ALS	77.8 %	99.6 %	99.0 %	99.6 %
HALS	48.8 %	92.0 %	98.4 %	96.6 %
DCPD-SeROAP	63.8 %	97.8 %	99.4 %	99.6 %
DCPD-ST-HOSVD	67.0 %	98.0 %	99.6 %	99.8 %
DCPD-THOSVD	70.0 %	99.0 %	99.6 %	99.8 %

Table 3.3: Percentage of tensors in which the exact CP decomposition is succeeded.

- The performance of the DCPD algorithm reinforces Conjecture 3.2.8 [4];
- The scenario $3 \times 3 \times 3$ presents the poorer performance because the algorithms do not satisfy the residual condition $\|\boldsymbol{\mathcal{E}}\| \leq 10^{-6}$ before the maximum number of iterations is attained. As a metter of fact, when the rank is of the same order of the dimensions, these algorithms need more iterations to converge.

3.3.3 Residual vs iteration

Figure 3.2 presents the performance of the algorithms in terms of the average of the normalized norm of residuals per iteration, namely $\mathbb{E}\{\|\boldsymbol{\mathcal{E}}[R,l]\|/\|\boldsymbol{\mathcal{T}}\|\}$, for different values of the signal-to-noise (SNR) ratio for $5 \times 5 \times 5$- rank-3 tensors. Again, a sample of 500 complex tensors with real and imaginary parts of the entries uniformly distributed in $[-1,1]$ is considered. Both ALS and HALS algorithms are randomly initialized. The same stopping criteria of Section 3.3.2 is established here. Additive Gaussian noise is considered in our simulations.

We note in this figure that average residual decreases more quickly for DCPD-SeROAP and DCPD-STHOSVD when compared to the other algorithms. For lower SNRs, most of the algorithms need a few iterations to converge to some stationary value. Notice that HALS takes more iterations to reduce the residual in all scenarios. For DCPD-THOSVD, we see that for an SNR of $0\ dB$ and the iteration greater than 5, the curve lies above the other ones. The reason of that is because for most of tensors the residual $\|\boldsymbol{\mathcal{E}}[R,l]\|$ does not attain its minimal value before the maximal number of iterations is attained. This fact is more clear explained in the next section.

3.3.4 Convergence vs Iteration

Let χ_l be the proportion of $5 \times 5 \times 5$- rank-3 tensors over all 500 ones at which the simulation has finished before l iterations. For the same noisy scenarios evaluated in the previous section, we present in the figure below the metric $\chi_l,\ 1 \leq l \leq 1000$ for all aforementioned low rank-1 approximation algorithms.

Figure 3.3 shows that ALS performs better than the other algorithm in all scenarios. In the scenario with SNR of 10 dB, while approximately 60% of simulations finished up to 100 iterations for ALS,

[4] See the explanation in next bulet to understand why the scenario $3 \times 3 \times 3$ cannot be used to infer that our conjecture fails.

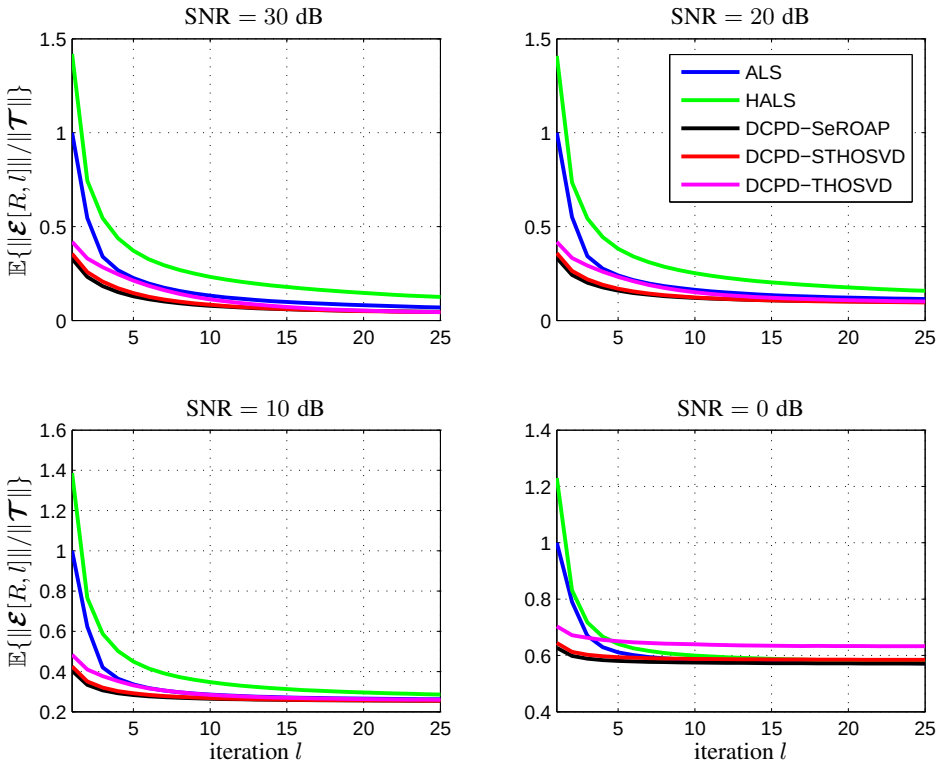


Figure 3.2: Mean of $\|\boldsymbol{\mathcal{E}}[R,l]\|/\|\boldsymbol{\mathcal{T}}\|$ for different values of SNR.

only 40% finished before the same number of iterations for the other algorithms. Notice also that 80% of simulations end up to 500 iterations with ALS, and approximately 65% with HALS and DCPD-SeROAP for the scenario with 0 dB. Among the iterative deflation algorithms, DCPD-SeROAP is the one that presents the more satisfactory performance. Notice that approximately along all scenarios, DCPD-SeROAP is at least as good as the other deflation methods. Apart from the last scenario, the HALS algorithm presents the worst performance. Yet, for the scenario with 0 dB, we note that the algorithms do not converge for some tensors. For instance, less than 60% of simulations converged before the maximum number of iterations for DCPD-THOSVD algorithm, which can explain the reason why the norm of the residual does not decrease in average for the same value obtained with the other algorithms, as described in Figure 3.2.

3.3.5 Residual vs rank

Now, we compare the algorithms for three SNR scenarios by computing $\mu = \mathbb{E}\{\|\boldsymbol{\mathcal{E}}[R, l_{max}]\|/\|\boldsymbol{\mathcal{T}}\|\}$, where l_{max} is the maximum iteration in which the stopping criterion is satisfied (or the last computed normalized residual). We perform simulations with different ranks for $5 \times 5 \times 5$ complex tensors. The results are shown in Figure 3.4.

In the first scenario (SNR $=$ 40 dB), we note that when the rank increases, the residual μ also increases. In this scenario, HALS presents the higher residual for all rank values. For the middle scenario (SNR $=$ 20 dB), the performance of all algorithms is approximately the same for all ranks,

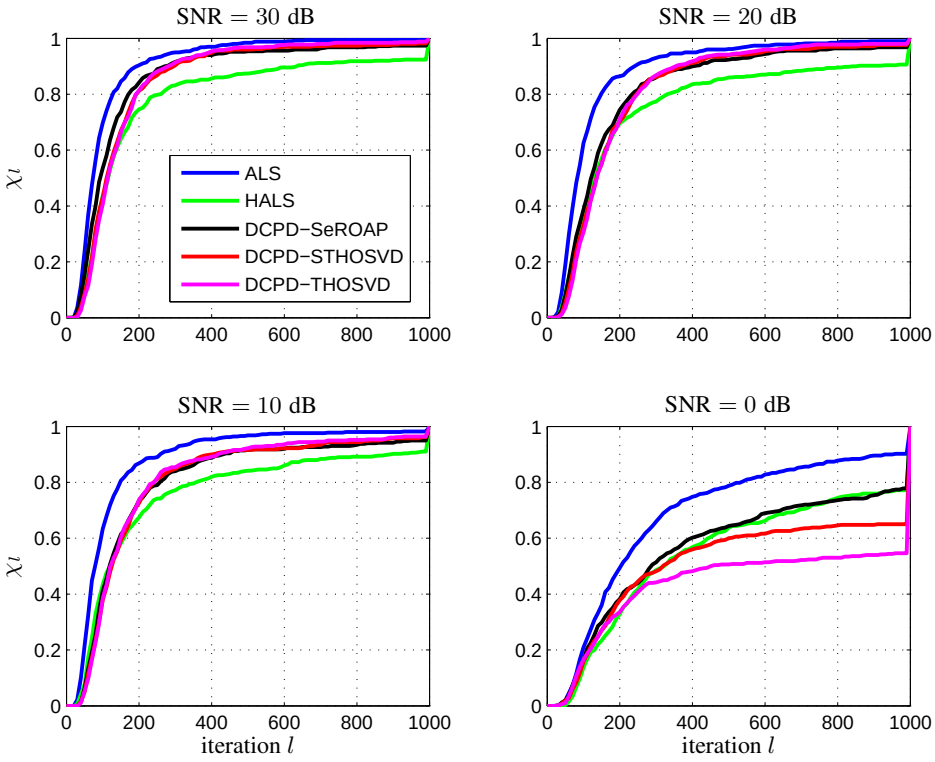


Figure 3.3: Percentage of tensors at which the simulations has finished up to l iterations.

with the residual μ in HALS slightly larger than those of the other algorithms. In the last scenario with lower SNR, the behavior of μ with the rank variation is the opposite of the first scenario. As a metter of fact, the higher is the rank, smaller is the residual μ. The figure also shows that all algorithms present approximately the same performance, except to DCPD-THOSVD, which is slightly worst.

3.3.6 Residual vs order

Figure 3.5 shows the average normalized residual μ under tensor order variation. We evaluate the performance of the algorithms under a noiseless scenario, and under scenarios with SNR $= 20$ dB and SNR $= 0$ dB. A sample of 500 complex tensors with dimensions $I_1 = I_2 = \cdots = I_N = 3$ and rank $R = 3$ is generated according to a uniform distribution as before.

As we discussed in Chapter 2, the approximation error of SeROAP algorithm is degraded for higher order tensors. This drawback is reflected into DCPD-SeROAP algorithm, where we clearly note that it does not work for $N \geq 5$, manly for high SNR (or noiseless scenario). DCPD-THOSVD also presents a poor performance, manly for the first scenario. On the other hand, DCPD-STHOSVD is robust to order variation, as ALS and HALS are. The latter, however, presents a slight increase of the residual for 6-th order tensors.

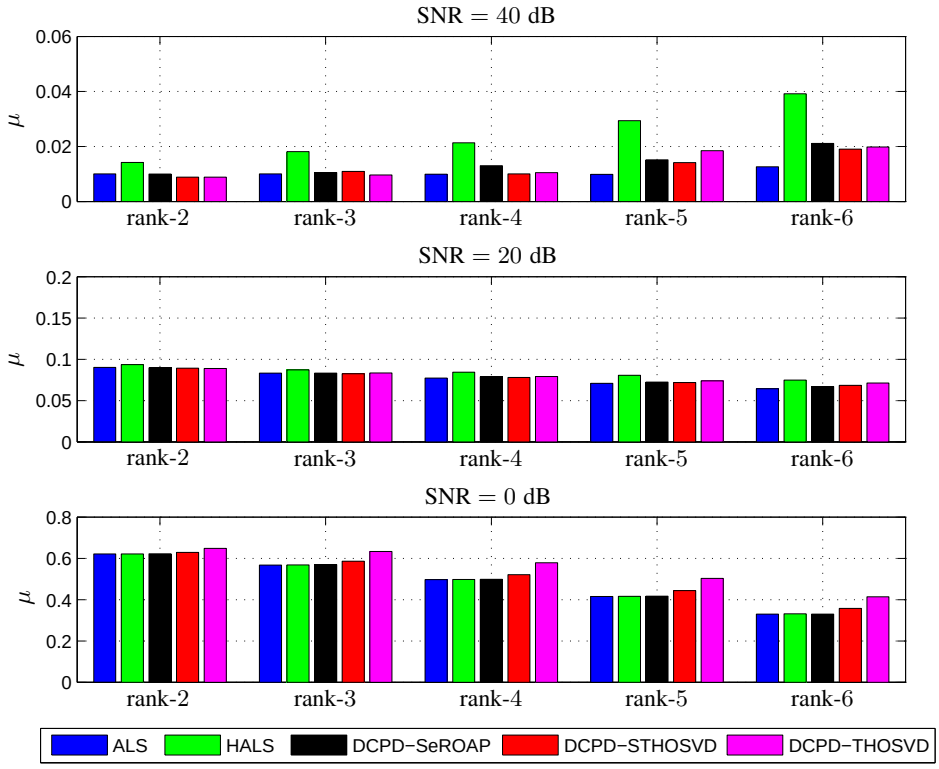


Figure 3.4: Mean of the normalized residual μ for different ranks and $5 \times 5 \times 5$ complex tensors.

3.4 Chapter Summary and Directions

We point out the following conclusions and future directions about iterative deflation:

- *DCPD algorithm.* we proposed an iterative deflation algorithm, called DCPD, based on finite rank-1 approximation algorithms to compute de CP decomposition and low rank tensor approximations. Instead of using an alternating update of factors for the rank-1 components as it is performed in HALS algorithm, we used the following finite rank-1 approximations: SeROAP, THOSVD and ST-HOSVD. The idea was the fast reduction of residuals in a few iterations, compared to standard methods like ALS and HALS itself. We have cogitated to employ an iterative algorithm to compute the best rank-1 approximation as the update of the rank-1 components in DCPD, in oder to reduce more quickly the residuals. However, the complexity per iteration would be unbounded in that case, so we did not implemented such an algorithm.

- *Theoretical study on iterative deflation.* We have showed in Corollary 3.2.3 that the norm of residuals $\|\boldsymbol{\mathcal{E}}[r, l]\|$ is a monotonic decreasing sequence on r and l when the rank-1 components in DCPD algorithm are updated with the best rank-1 approximation. In Proposition 3.2.5 , we have proved that our iterative deflation method can recover an exact CP tensor decomposition if the residuals do not fall within an arbitrary small cone. Another result was the reduction of residuals with high probability when a defined probability function of residuals is continuous in some interval. This was showed in Proposition 3.2.7. Finally, we conjectured that there exist

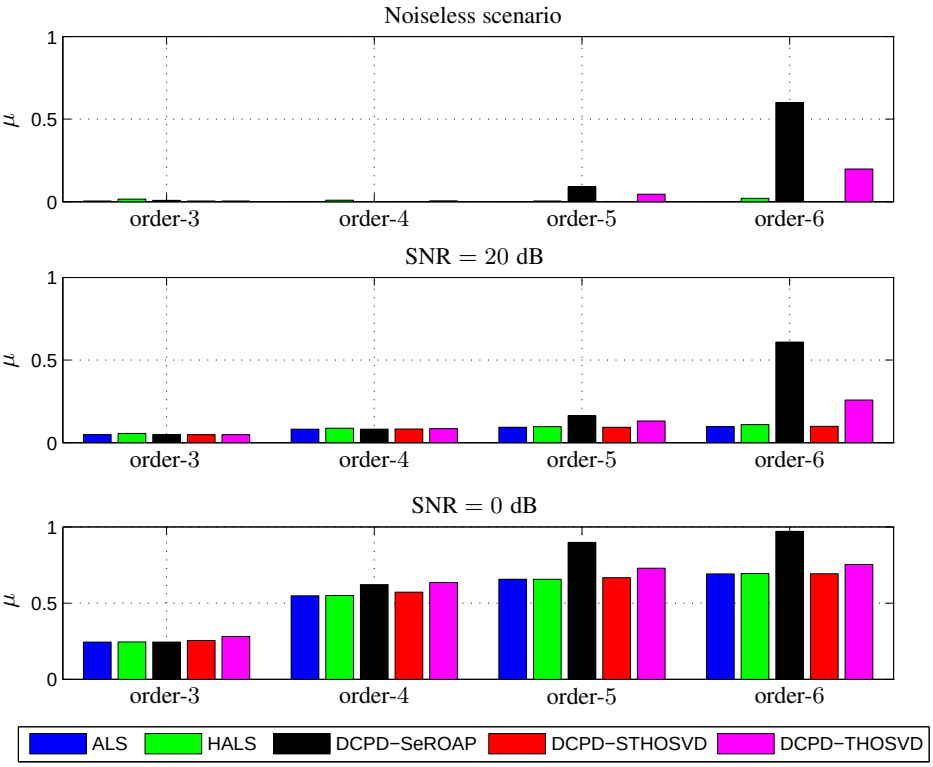


Figure 3.5: Mean of the normalized residual for different orders and $I_1 = I_2 = \cdots I_N = 3$.

absolute continuous probability measures for the entries of tensors with rank at most R such that the DCDP algorithm, with a best rank-1 approximation as the procedure to update the rank-1 tensors $\boldsymbol{\mathcal{X}}[r, l]$, converges to an exact rank R CP decomposition with very high probability (see Conjecture 3.2.8). Hereafter, we intend to work on the proof of our conjecture. Some tools on algebraic geometry and measure theory may be starting points to tackle this problem.

- *Performance of iterative deflation.* We have evaluated five tensor algorithms on several fronts: complexity per iteration, percentage of successful decompositions, convergence versus iteration, residual versus iteration, residual versus rank, and residual versus order. Based on our experiments, we establish a subjective qualification factor from 1 to 5 for each of the aforementioned performance metrics, denoting the ranking of the algorithms. We set the value 5 for the algorithm considered the one with poorer performance for the studied scenarios, and 1 for the best one. This is summarized in the table below.

Algorithm	complexity per iteration	percentage of successful decompositions	convergence versus iteration	residual versus iteration	residual versus rank	residual versus order
ALS	1	1	1	3	1	1
HALS	2	5	5	5	5	3
DCPD-SeROAP	4	4	2	1	2	5
DCPD-STHOSVD	3	3	3	2	3	2
DCPD-THOSVD	5	2	4	4	4	4

Table 3.4: Summary of performance factor for the algorithms.

Table 3.4 shows that ALS was the best algorithm whereas HALS the worst one in general for the evaluated scenarios. The DCPD-SeROAP algorithm is considered a robust algorithm in terms of convergence and residual versus iteration, and also in terms of rank variation. However, we saw that it is a very bad algorithm for tackling higher order tensors. Although it was considered the fourth algorithm in terms of percentage of successful decompositions, the performance was very close to ALS (in first position) for higher dimension scenarios (see table 3.2). As HALS, DCPD-THOSVD did not present good results. Finally, DCPD-STHOSVD had a moderate performance.

CHAPTER 4

Multivariate quadratic systems and the best rank-1 tensor approximation

In this chapter we propose a new approach based on rank-1 three-way tensor approximations that extracts solutions of multivariate quadratic polynomial systems in real and complex fields. This approach allows us to make use of standard iterative tensor algorithms [Comon 2009b], which in practice present satisfactory performances in time and convergence.

We start the chapter by presenting a brief introduction on the existing mathematical tools and some applications on polynomial optimization theory, in particular we focus on multivariate quadratic systems. The following sections deal with the formulation, discussion and experiments of our rank-1 tensor approach to tackle this problem.

First, we show the details of reducing a general multivariate quadratic system in the real field into a best rank-1 approximation problem. This is proved along four propositions describing the reduction (or equivalence) between intermediate problems. Our rank-1 tensor approach is discussed in the following section, where the advantages and drawbacks are presented.

Second, we extend our approach to the complex case in which only quadratic monomials of the form $x_i^* x_j,\ 1 \leq i, j \leq 2m$, where m is the number of equations, are allowed for. In that case, the system is reduced to a conjugated partially symmetric rank-1 tensor approximation. We omit details of the reduction since they follow the same procedures described in the real case. In the sequel, we propose an alternating algorithm to compute the best conjugated partially symmetric rank-1 approximation. This algorithm, called CPS, is based on the stationary equations of a least square problem.

In the parts about examples and performance of the rank-1 tensor approach, we extract solutions of a multivariate quadratic system using ALS for real tensors, and CPS for complex tensors. We present an example of an over-determined real system with and without perturbation, where the solution is obtained by using different initializations methods. We also present an example of computing a solution of an under-determined real system. The last example is a complex over-determined system with a single complex solution that is extracted by CPS algorithm. We also perform a set of Monte Carlo simulations for several scenarios in real and complex case, where we show the computational time and the approximation error of estimations, in order to evince the usefulness of our approach.

Finally, we close this chapter drawing some conclusions and giving some directions to future work.

Main contributions of this chapter:

- *Reduction of multivariate quadratic systems to best rank-1 tensor approximations.* The best rank-1 tensor approximation is an optimization problem than can be applied to solve multivariate polynomial quadratic systems. We proof that a general quadratic system in the real field can be reduced to a best rank-1 approximation problem of a three-way real tensor. This tensor is constructed from the parameters of the system. The extension to specific complex systems is also done using the same approach. In this case, we show that the system is reduced to a conjugated partially symmetric rank-1 three-way tensor approximation problem, which can be tackled by an alternating algorithm.

- *Rank-1 approximation algorithm.* We propose an algorithm to compute the (conjugated) partially symmetric rank-1 three-way tensor approximation problem. This algorithm, called CPS, is developed from the stationary equations of the least squares problem $min_{\boldsymbol{y},\boldsymbol{w}} \ \|\boldsymbol{\mathcal{T}} - \boldsymbol{y} \otimes \boldsymbol{y}^* \otimes \boldsymbol{w}\|^2$.

- *Performance of the rank-1 tensor approach.* ALS and CPS algorithms present satisfactory performance to extract a solution of a quadratic system in a reasonable time (e.g. a few seconds for a real square system of 10 variables/equations). We draw some examples and perform Monte Carlo experiments to evince the efficiency of our approach for real and complex systems.

Contents

4.1 An introduction to multivariate quadratic systems

Multivariate polynomial optimization (MPO) is still an important topic in mathematics wherein a vast amount of applications takes place. Formally, a MPO problem consists in minimizing a polynomial cost function constrained by a set of polynomial equations and/or inequalities. In the last decades, some mathematical techniques have been applied to solve this problem, such as Gröbner bases, resultants and eigenvalues/eigenvectors of companion matrices [Cox 2006], semidefinite relaxations [Parrilo 2003, Lasserre 2001, Bucero 2014], and numerical homotopy [Li 1997, Verschelde 1999].

The complexity of a general polynomial optimization problem is NP-hard, even for quadratic polynomials with affine constraints [Pardalos 1991]. As a metter of fact, any MPO problem can be modified to a quadratic optimization (QO) problem by adding some variables and constraints.

A particular problem in this vein is the multivariate quadratic (MQ) polynomial systems, which are of great interest in various areas. In game theory, the Nash equilibria of a non-cooperative game between two players can be found by solving an MQ system [Lipton 2004]. In cryptography, the security of systems depends on the difficult to solve large quadratic systems in finite fields [Courtois 2000, Thomae 2012]. In [Lebrun 2004, Lin 2004], the authors present the design of multivariate filter banks modeled by quadratic polynomial systems. A last application lies in multilinear algebra [Brachat 2009], where the authors propose an efficient algorithm to decompose a symmetric tensor and show the equivalence between an existence condition of the decomposition and the solution of an MQ system.

Besides the techniques mentioned in the first paragraph, other methods can be employed to solve multivariate quadratique systems. For instance, we cite Newton-based and tensor-based[1] algorithms [Dennis Jr 1996, Schnabel 1984], low-rank matrix recovery strategy [Davenport 2016], and symbolic computation [Grigoriev 2005].

The goal of this chapter is to present a rank-1 tensor approach to solve general MQ systems in the real field, and also to solve a particular extension in complex fields. In the real case, we show that a general MQ system can be reduced to a best rank-1 three-way tensor approximation problem [Lathauwer 2000, Friedland 2013, da Silva 2014], while the particular complex system can be reduced to a conjugated partially symmetric rank-1 three-way approximation. Despite the fact that most rank-1 tensor approximation algorithms do not guarantee to yield the optimum solution of the problem, it is known that standard algorithms, such as alternating least squares (ALS) and gradient-based methods [Comon 2009b] generally give satisfactory solutions in practice. Reducing the MQ problem to a rank-1 approximation can hence be attractive to compute a solution of MQ systems.

To compute a conjugated partially symmetric rank-1 approximation of some complex tensor $\mathcal{T}$, we propose an alternating algorithm, called CPS. Some examples and experiments confirm the efficiency of rank-1 approximation methods to extract solutions of MQ systems.

[1] Tensor-based and rank-1 tensor algorithms are not the same techniques.

4.2 From quadratic systems to best rank-1 approximations

Let P1 be the general system of quadratic polynomial equations given by

$$\textbf{P1:} \begin{cases} \boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{b}_1^{\mathsf{T}}\boldsymbol{x} + c_1 = 0 \\ \qquad\qquad \vdots \\ \boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}_m\boldsymbol{x} + \boldsymbol{b}_m^{\mathsf{T}}\boldsymbol{x} + c_m = 0, \end{cases}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{A}_j \in \mathbb{R}^{n\times n}$ are symmetric matrices, $\boldsymbol{b}_j \in \mathbb{R}^n$ and $c_j \in \mathbb{R}$, $1 \leq j \leq m$.

Let also $\mathcal{I}$ be the ideal in $\mathbb{R}[\boldsymbol{x}]$ generated by the set of polynomials of (P1), that is,

$$\mathcal{I} = \langle h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})\rangle,$$

where $h_j(\boldsymbol{x}) = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}_j\boldsymbol{x} + \boldsymbol{b}_j^{\mathsf{T}}\boldsymbol{x} + c_j, 1 \leq j \leq m$. Thus, the set of solutions of (P1) is defined by the affine variety $V(\mathcal{I}) := \{\boldsymbol{x} \in \mathbb{R}^n \mid f(\boldsymbol{x}) = 0,\ \forall f \in \mathcal{I}\}$. Also define the optimization problem (P2) as follows

$$\textbf{P2:} \begin{cases} p_2^\star = \min\limits_{\boldsymbol{y}} p(\boldsymbol{y}) \\ \\ s.t.\ \|\boldsymbol{y}\| = 1, \end{cases}$$

where $p(\boldsymbol{y}) = \sum\limits_{j=1}^{m} (\boldsymbol{y}^{\mathsf{T}}\boldsymbol{Q}_j\boldsymbol{y})^2$, for $\boldsymbol{y} \in \mathbb{R}^{n+1}$ and

$$\boldsymbol{Q}_j = \begin{bmatrix} \boldsymbol{A}_j & \boldsymbol{b}_j/2 \\ \\ \boldsymbol{b}_j^{\mathsf{T}}/2 & c_j \end{bmatrix} \in \mathbb{R}^{n+1\times n+1}.$$

Define the set of global minimizers of (P2) as $\mathcal{S}_{P2} = \{\boldsymbol{y} \in \mathbb{R}^{n+1} \mid p(\boldsymbol{y}) = p_2^\star, \|\boldsymbol{y}\| = 1\}$, and the subset $\bar{\mathcal{S}}_{P2} \subseteq \mathcal{S}_{P2}$ given by $\bar{\mathcal{S}}_{P2} = \mathcal{S}_{P2} \cap (\mathbb{R}^n \times \mathbb{R}\backslash\{0\})$. In other words, $\bar{\mathcal{S}}_{P2}$ is the set of solutions of Problem (P2) such that $y_{n+1} \neq 0$. Let also $\mathcal{N} = \{\boldsymbol{z} \in \mathbb{R}^{n+1} \mid \boldsymbol{z} = \boldsymbol{y}/y_{n+1},\ \forall \boldsymbol{y} \in \bar{\mathcal{S}}_{P2}\}$. Proposition below connects Problems (P1) and (P2).

Proposition 4.2.1 *If $V(\mathcal{I}) \neq \emptyset$ then $V(\mathcal{I}) \times \{1\} = \mathcal{N}$.*

Proof: By setting $\boldsymbol{y} = [\boldsymbol{x}\ 1]^{\mathsf{T}}$, it turns out that

$$\boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}_j\boldsymbol{x} + \boldsymbol{b}_j^{\mathsf{T}}\boldsymbol{x} + c_j = \boldsymbol{y}^{\mathsf{T}}\boldsymbol{Q}_j\boldsymbol{y},$$

$\forall j \in \{1, 2, \ldots, m\}$. This shows that the set of solutions of the following system, equivalent to (P1),

$$\begin{cases} \boldsymbol{y}^{\mathsf{T}}\boldsymbol{Q}_j\boldsymbol{y} = 0, \text{ for } 1 \leq j \leq m \\ y_{n+1} = 1 \end{cases} \tag{4.1}$$

is $V(\mathcal{I}) \times \{1\}$. Now, consider the optimization problem

$$\begin{aligned} p_2^\star &= \min_{\boldsymbol{y}} p(\boldsymbol{y}) \\ s.t.\ &\|\boldsymbol{y}\| = 1. \end{aligned} \tag{4.2}$$

Since $p(\boldsymbol{y})$ is by construction the sum of squares of the quadratic polynomials $\boldsymbol{y}^\mathsf{T}\boldsymbol{Q}_j\boldsymbol{y},\ 1 \leq j \leq m$, and $V(\mathcal{I}) \neq \emptyset$, it follows that $p_2^\star = 0$. Thus, $\forall \bar{\boldsymbol{y}} \in V(\mathcal{I}) \times \{1\}$, $p(\bar{\boldsymbol{y}}/\|\bar{\boldsymbol{y}}\|) = 0$, which implies that $\bar{\boldsymbol{y}}/\|\bar{\boldsymbol{y}}\| \in \bar{\mathcal{S}}_{P2}$ and $\bar{\boldsymbol{y}} \in \mathcal{N}$. This proves that $V(\mathcal{I}) \times \{1\} \subseteq \mathcal{N}$.

On the other hand, $\forall \boldsymbol{y} \in \bar{\mathcal{S}}_{P2}$, $p(\boldsymbol{y}/y_{n+1}) = 0$, which implies that $(\boldsymbol{y}/y_{n+1})^\mathsf{T}\boldsymbol{Q}_j(\boldsymbol{y}/y_{n+1}) = 0, 1 \leq j \leq m$. Thus, $\boldsymbol{y}/y_{n+1}$ is solution of (4.1), which leads to $\mathcal{N} \subseteq V(\mathcal{I}) \times \{1\}$, and the proof is complete. □

Proposition 4.2.1 says that all solutions of System (P1) can be recovered from Problem (P2) by taking the n first elements of $\boldsymbol{y}/y_{n+1},\ \forall \boldsymbol{y} \in \bar{\mathcal{S}}_{P2}$.

Remark 1 *When $\boldsymbol{b}_j = \boldsymbol{0}, c_j = 0,\ \forall j \in \{1, 2, \ldots, m\}$, $V(\mathcal{I})\backslash\{\boldsymbol{0}\}$ can be viewed as a projective variety, since $\boldsymbol{x} \in V(\mathcal{I}) \implies \alpha\boldsymbol{x} \in V(\mathcal{I})\ \forall \alpha \in \mathbb{R}$. Therefore, it is enough to solve the system for $\|\boldsymbol{x}\| = 1$. By setting $\boldsymbol{y} = \boldsymbol{x}$, and $\boldsymbol{Q}_j = \boldsymbol{A}_j, 1 \leq j \leq m$, all nontrivial solutions of (P1) can be recovered by solving (P2).*

Now, let the following optimization problem

$$\textbf{P3:} \begin{cases} p_3^\star = \max\limits_{\boldsymbol{y}} (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}\boldsymbol{M}(\boldsymbol{y} \boxtimes \boldsymbol{y}) \\ \\ s.t.\ \|\boldsymbol{y}\| = 1, \end{cases}$$

where $\boldsymbol{M} \in \mathbb{R}^{(n+1)^2 \times (n+1)^2}$ is a semidefinite positive matrix given by $\boldsymbol{M} = \lambda_{max}\boldsymbol{I} - \boldsymbol{M}_Q$, where $\boldsymbol{M}_Q = \sum_{j=1}^m \boldsymbol{Q}_j \boxtimes \boldsymbol{Q}_j$, $\boldsymbol{I}$ is the identity matrix, and λ_{max} is the largest eigenvalue of $\boldsymbol{M}_Q$.

Proposition 4.2.2 *(P2) $\iff$ (P3).*

Proof: We have

$$p(\boldsymbol{y}) = \sum_{j=1}^m (\boldsymbol{y}^\mathsf{T}\boldsymbol{Q}_j\boldsymbol{y})^2 = \sum_{j=1}^m (\boldsymbol{y}^\mathsf{T}\boldsymbol{Q}_j\boldsymbol{y}) \boxtimes (\boldsymbol{y}^\mathsf{T}\boldsymbol{Q}_j\boldsymbol{y}) = \sum_{j=1}^m (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}(\boldsymbol{Q}_j \boxtimes \boldsymbol{Q}_j)(\boldsymbol{y} \boxtimes \boldsymbol{y})$$
$$= (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T} \left(\sum_{j=1}^m \boldsymbol{Q}_j \boxtimes \boldsymbol{Q}_j \right) (\boldsymbol{y} \boxtimes \boldsymbol{y}) = (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}\boldsymbol{M}_Q(\boldsymbol{y} \boxtimes \boldsymbol{y}).$$

Hence, it follows that

$$\min_{\|\boldsymbol{y}\|=1} (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}\boldsymbol{M}_Q(\boldsymbol{y} \boxtimes \boldsymbol{y}) = \max_{\|\boldsymbol{y}\|=1} -(\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}\boldsymbol{M}_Q(\boldsymbol{y} \boxtimes \boldsymbol{y}) \iff \max_{\|\boldsymbol{y}\|=1} \lambda_{max} - (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}\boldsymbol{M}_Q(\boldsymbol{y} \boxtimes \boldsymbol{y})$$
$$= \max_{\|\boldsymbol{y}\|=1} \lambda_{max}(\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}(\boldsymbol{y} \boxtimes \boldsymbol{y}) - (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}\boldsymbol{M}_Q(\boldsymbol{y} \boxtimes \boldsymbol{y}) = \max_{\|\boldsymbol{y}\|=1} (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}(\lambda_{max}\boldsymbol{I} - \boldsymbol{M}_Q)(\boldsymbol{y} \boxtimes \boldsymbol{y})$$
$$= \max_{\|\boldsymbol{y}\|=1} (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T}\boldsymbol{M}(\boldsymbol{y} \boxtimes \boldsymbol{y}).$$

□

Remark 2 *Problem (P3) can be viewed as a symmetric best rank-1 approximation of a fourth order tensor. Indeed, it can be written as $\min\limits_{\|\boldsymbol{y}\|=1} \|\boldsymbol{\mathcal{M}} - \boldsymbol{y} \otimes \boldsymbol{y} \otimes \boldsymbol{y} \otimes \boldsymbol{y}\|^2$, where $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{n+1 \times n+1 \times n+1 \times n+1}$*

is a symmetric tensor constructed from $\boldsymbol{M}$. This problem is difficult to solve and some standard algorithms, such as ALS, could be adapted to deal with it. However, ignoring the symmetry generally leads to faster convergence. For this reason, reducing the system to a standard three-way rank-1 approximation problem is attractive under the algorithmic point of view.

In the following, we establish a link between (P3) and a rank-1 three-way tensor approximation problem. Let $(\lambda_k, \boldsymbol{v}_k),\ 1 \leq k \leq K,\ K = \operatorname{rank} \boldsymbol{M}$, be the k-th eigenpair of $\boldsymbol{M}$. Due to the semi-definiteness of $\boldsymbol{M}$, we can define real vectors $\boldsymbol{t}_k = \sqrt{\lambda_k}\boldsymbol{v}_k \in \mathbb{R}^{(n+1)^2}$. Set $\boldsymbol{T}_k = \operatorname{Unvec}(\boldsymbol{t}_k)$. The matrices $\boldsymbol{T}_k$ can be viewed as frontal slices of a three-way tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n+1 \times n+1 \times K}$ [Kolda 2009], as illustrated in Figure 4.1.

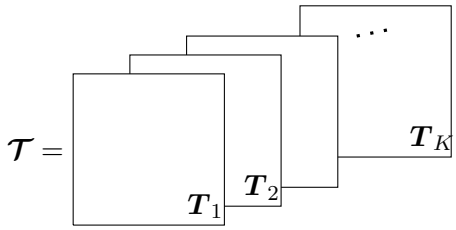


Figure 4.1: Frontal slices of a three-way tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n+1 \times n+1 \times K}$.

Now, define the following rank-1 tensor approximation problem with two identical factors.

$$\textbf{P4:} \begin{cases} p_4^\star = \min\limits_{\boldsymbol{y},\boldsymbol{w}} \|\boldsymbol{\mathcal{T}} - \boldsymbol{y} \otimes \boldsymbol{y} \otimes \boldsymbol{w}\|. \\ \\ s.t.\ \|\boldsymbol{y}\| = 1. \end{cases}$$

The following proposition holds.

Proposition 4.2.3 *(P3) $\Longleftrightarrow$ (P4).*

Proof: Expand matrix $\boldsymbol{M}$ into its eigencomponents as: $\boldsymbol{M} = \sum_{k=1}^{K} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^\mathsf{T} = \sum_{k=1}^{K} \boldsymbol{t}_k \boldsymbol{t}_k^\mathsf{T}$, and write the objective function of (P3) as $(\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T} \boldsymbol{M} (\boldsymbol{y} \boxtimes \boldsymbol{y}) = \sum_{k=1}^{K} (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T} \boldsymbol{t}_k \boldsymbol{t}_k^\mathsf{T} (\boldsymbol{y} \boxtimes \boldsymbol{y})$. Set $w_k = (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T} \boldsymbol{t}_k$. Problem (P3) can be rewritten as

$$\max_{(\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}} \textstyle\sum_{k=1}^{K} w_k \boldsymbol{t}_k^\mathsf{T} (\boldsymbol{y} \boxtimes \boldsymbol{y}) \tag{4.3}$$

where $\mathcal{C} = \{(\boldsymbol{y}, \boldsymbol{w}) \in \mathbb{R}^n \times \mathbb{R}^K, :\ \|\boldsymbol{y}\| = 1,\ \boldsymbol{w} = [w_1\ w_2\ \ldots\ w_K]^\mathsf{T},\ w_k = (\boldsymbol{y} \boxtimes \boldsymbol{y})^\mathsf{T} \boldsymbol{t}_k,\ 1 \leq k \leq K\}$. Next, we have the equivalence

$$\max_{(\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}} \sum_{k=1}^{K} w_k \boldsymbol{t}_k^\mathsf{T} (\boldsymbol{y} \boxtimes \boldsymbol{y}) \iff \min_{(\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}} \sum_{k=1}^{K} \boldsymbol{t}_k^\mathsf{T} \boldsymbol{t}_k - w_k \boldsymbol{t}_k^\mathsf{T} (\boldsymbol{y} \boxtimes \boldsymbol{y}).$$

Yet, since $(\boldsymbol{y}, \boldsymbol{w}) \in \mathcal{C}$, it turns out that $\boldsymbol{t}_k^\mathsf{T} \boldsymbol{t}_k - w_k \boldsymbol{t}_k^\mathsf{T} (\boldsymbol{y} \boxtimes \boldsymbol{y}) = \|\boldsymbol{t}_k - w_k (\boldsymbol{y} \boxtimes \boldsymbol{y})\|^2,\ 1 \leq k \leq K$. Therefore, Problem (P3) is equivalent to

$$\min_{(\boldsymbol{y},\boldsymbol{w}) \in \mathcal{C}} \sum_{k=1}^{K} \|\boldsymbol{t}_k - w_k (\boldsymbol{y} \boxtimes \boldsymbol{y})\|^2 \tag{4.4}$$

or, to minimize the Lagrangian:

$$\Upsilon(\boldsymbol{y},\boldsymbol{w}) = \sum_{k=1}^{K} \|\boldsymbol{t}_k - w_k(\boldsymbol{y}\boxtimes\boldsymbol{y})\|^2 + \eta(\|\boldsymbol{y}\|-1) + \sum_{k=1}^{K}\gamma_k(w_k - (\boldsymbol{y}\boxtimes\boldsymbol{y})^{\mathsf{T}}\boldsymbol{t}_k),$$

where $\gamma_k,\ 1\leq k\leq K$, and η are Lagrangian multipliers.

We now need to relax the constraint on w_k. Stationary points w.r.t. w_k yield

$$\frac{\partial\Upsilon(\boldsymbol{y},\boldsymbol{w})}{\partial w_k} = -2(\boldsymbol{t}_k - w_k(\boldsymbol{y}\boxtimes\boldsymbol{y}))^{\mathsf{T}}(\boldsymbol{y}\boxtimes\boldsymbol{y}) + \gamma_k = 0 \implies$$
$$\implies (-\boldsymbol{t}_k + w_k(\boldsymbol{y}\boxtimes\boldsymbol{y}))^{\mathsf{T}}(\boldsymbol{y}\boxtimes\boldsymbol{y}) + \gamma_k/2 = 0.$$

Now using all the constraints defined in $\mathcal{C}$ (*i.e.* stationary equations w.r.t. γ_k and η), the latter equation leads to $\gamma_k = 0$, $\forall k$, which means that the constraints on w_k do not need to be imposed. Hence we end up with the simplified Lagrangian:

$$\Upsilon(\boldsymbol{y},\boldsymbol{w}) = \sum_{k=1}^{K} \|\boldsymbol{t}_k - w_k(\boldsymbol{y}\boxtimes\boldsymbol{y})\|^2 + \eta(\|\boldsymbol{y}\|-1). \tag{4.5}$$

To complete the proof, note that by reshaping the entries of the vectors $\boldsymbol{t}_k$ using the Unvec operator, (5) can still be written with the help of matrix slices:

$$\min_{\|\boldsymbol{y}\|=1,\boldsymbol{w}} \textstyle\sum_{k=1}^{K} \|\boldsymbol{T}_k - w_k\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\|^2 \tag{4.6}$$

or in its tensor representation

$$\min_{\|\boldsymbol{y}\|=1,\boldsymbol{w}} \|\boldsymbol{\mathcal{T}} - \boldsymbol{y}\otimes\boldsymbol{y}\otimes\boldsymbol{w}\|^2.$$

□

Problem (P4) is a best partially symmetric rank-1 approximation, which cannot be directly tackled by standard tensor approximation algorithms. However, this problem can still be changed in order to drop the constraint of identical factors.

According to [Friedland 2013], the best rank-1 approximation of a real symmetric tensor with respect to a subset of indices (modes) can be chosen symmetric with respect to this subset (see Theorem 1 therein). Thus, if tensor $\boldsymbol{\mathcal{T}}$ in Problem (P4) were symmetric with respect to the first two modes, we would not need to impose any constraint on the factors, so that the best rank-1 approximation could be chosen partially symmetric.

In general, $\boldsymbol{\mathcal{T}}$ is not symmetric with respect to the first two modes. However, the linear transformation $\boldsymbol{T}'_k = (\boldsymbol{T}_k + \boldsymbol{T}_k^{\mathsf{T}})/2$ on each k-th slice $\boldsymbol{T}_k$ of $\boldsymbol{\mathcal{T}}$, ensures that any best partially symmetric rank-1 approximation of $\boldsymbol{\mathcal{T}}$ is actually a best rank-1 approximation of a tensor $\boldsymbol{\mathcal{T}}'$ with slices $\boldsymbol{T}'_k$ (which is therefore symmetric with respect to the first two modes). In other words, Problem (P4) can be reduced to the problem

$$\textbf{P5:}\ \begin{cases} p_5^\star = \min\limits_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{w}} \|\boldsymbol{\mathcal{T}}' - \boldsymbol{x}\otimes\boldsymbol{y}\otimes\boldsymbol{w}\|. \\ \\ s.t.\ \|\boldsymbol{y}\| = 1. \end{cases}$$

Before proving this, we need the following lemma.

Lemma 4.2.4 *Let* $f(\boldsymbol{X},\boldsymbol{\alpha}) = \sum_{k=1}^{K} \|\boldsymbol{A}_k - \alpha_k \boldsymbol{X}\|^2$ *and* $g(\boldsymbol{X},\boldsymbol{\alpha}) = \sum_{k=1}^{K} \|(\boldsymbol{A}_k + \boldsymbol{A}_k^{\mathsf{T}})/2 - \alpha_k \boldsymbol{X}\|^2$ *with* $\boldsymbol{X}, \boldsymbol{A}_k \in \mathbb{R}^{n\times n}$, $1 \leq k \leq K$, $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \ldots\ \alpha_K]^{\mathsf{T}}$. *If* $\boldsymbol{X}$ *is symmetric, then*

$$\arg\min_{\boldsymbol{X},\boldsymbol{\alpha}} f(\boldsymbol{X},\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{X},\boldsymbol{\alpha}} g(\boldsymbol{X},\boldsymbol{\alpha}).$$

Proof: Let $h(\boldsymbol{X},\boldsymbol{\alpha}) = \sum_{k=1}^{K} \|\boldsymbol{A}_k^{\mathsf{T}} - \alpha_k \boldsymbol{X}\|^2$. Since $\boldsymbol{X}$ is symmetric, it follows that

$$\arg\min_{\boldsymbol{X},\boldsymbol{\alpha}} f(\boldsymbol{X},\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{X},\boldsymbol{\alpha}} h(\boldsymbol{X},\boldsymbol{\alpha}) = \arg\min_{\boldsymbol{X},\boldsymbol{\alpha}} f(\boldsymbol{X},\boldsymbol{\alpha}) + h(\boldsymbol{X},\boldsymbol{\alpha}).$$

By expanding the functions,

$$f(\boldsymbol{X},\boldsymbol{\alpha}) + h(\boldsymbol{X},\boldsymbol{\alpha}) = 2\|\boldsymbol{X}\|^2 \sum_{k=1}^{K} \alpha_k^2 - 2\sum_{k=1}^{K} \alpha_k \operatorname{trace}\{(\boldsymbol{A}_k + \boldsymbol{A}_k^{\mathsf{T}})\boldsymbol{X}\} + 2\sum_{k}^{K} \|\boldsymbol{A}_k\|^2, \text{ and}$$

$$g(\boldsymbol{X},\boldsymbol{\alpha}) = \|\boldsymbol{X}\|^2 \sum_{k=1}^{K} \alpha_k^2 - \sum_{k=1}^{K} \alpha_k \operatorname{trace}\{(\boldsymbol{A}_k + \boldsymbol{A}_k^{\mathsf{T}})\boldsymbol{X}\} + \frac{1}{4}\sum_{k=1}^{K} \|\boldsymbol{A}_k + \boldsymbol{A}_k^{\mathsf{T}}\|^2.$$

Thus, $f(\boldsymbol{X},\boldsymbol{\alpha}) + h(\boldsymbol{X},\boldsymbol{\alpha}) = 2g(\boldsymbol{X},\boldsymbol{\alpha}) + c$, for some constant c, and the proof is complete. □

Proposition 4.2.5 *Let* $\mathcal{S}_{P4}$ *and* $\mathcal{S}_{P5}$ *be the sets of rank-1 tensors that minimize Problems (P4) and (P5), respectively. Then* $\mathcal{S}_{P4} \subseteq \mathcal{S}_{P5}$.

Proof: Since $\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}$ is symmetric, we can apply Lemma 4.2.4 to the cost function of the slice form of Problem (P4) to show that

$$\arg\min_{\boldsymbol{y},\boldsymbol{w}} \sum_{k=1}^{K} \|\boldsymbol{T}_k - w_k \boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\|^2 = \arg\min_{\boldsymbol{y},\boldsymbol{w}} \sum_{k=1}^{K} \|\boldsymbol{T}'_k - w_k \boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\|^2.$$

That means that any rank-1 tensor $\boldsymbol{\mathcal{X}} = \boldsymbol{y} \otimes \boldsymbol{y} \otimes \boldsymbol{w} \in \mathcal{S}_{P4}$ is a partially symmetric rank-1 tensor that minimizes (P5), which is in turn a best rank-1 approximation from Theorem 1 in [Friedland 2013]. Therefore, $\forall \boldsymbol{\mathcal{X}} \in \mathcal{S}_{P4}, \boldsymbol{\mathcal{X}} \in \mathcal{S}_{P5} \implies \mathcal{S}_{P4} \subseteq \mathcal{S}_{P5}$. □

4.3 Discussion on the real tensor approach

The goal of this section is to point out the advantages and disadvantages of using rank-1 tensor approximation algorithms to solve quadratic polynomials systems in the real field. We start describing the drawbacks, but with the intention of emphasizing some aspects that mitigate part of the deficiencies of the approach.

- Apart from the case described in Remark 1, Problems (P1) and (P2) are not equivalent in general. Thus, solving Problem (P5) does not ensure that the factor $\boldsymbol{y}$ obtained with a rank-1 tensor approximation method satisfies the condition $y_{n+1} \neq 0$. In this case, the standard ALS algorithm can be adapted by setting $y_{n+1} = 1$ at every iteration in order to avoid undesirable solutions. However, this adaptation can compromise the convergence of the algorithm.

- Problems (P4) and (P5) are not equivalent either, which means that a non symmetric rank-1 approximation can be a minimizer of (P5). However, routine simulations have shown that ALS delivers (partially) symmetric rank-1 approximations for (partially) symmetric tensors.

- The complexity for constructing tensor $\boldsymbol{\mathcal{T}}$ is of order $\mathcal{O}\{mn^4 + n^6\}$, which comes mainly from the computation of $\boldsymbol{M}$ and the eigenpairs $(\lambda_k, \boldsymbol{v}_k), 1 \leq k \leq K$.

- Solutions delivered by iterative tensor approximation algorithms have an initialization dependence, which means that such methods do not guarantee that the optimum solution of the best rank-1 approximation problem is attained. However, the solution delivered by Problem (P5) allows us to test whether $\boldsymbol{x}$ satisfies or not System (P1). If it does not, we can run again the algorithms with other initializations.

- Contrary to (numerical) algebraic geometry methods, such as those described in [Cox 2006, Verschelde 1999], tensor approximation algorithms can find only a solution of (P1). As a metter of fact, the latter should run several times with different initializations in order to find other possible solutions.

On the other hand, we can take advantage of tensor approach for solving quadratic polynomial systems. We emphasize the following points:

- Numerical experiments for generic systems have shown that $y_{n+1} \neq 0$ is always satisfied for the standard ALS method with random initializations.

- The best rank-1 tensor approximation problem is well-posed, which means that Problem (P5) has always a partially symmetric minimizer.

- When System (P1) is inconsistent, the tensor approach gives a solution that fits (P1). Thus, tensor approach can be useful when a system becomes inconsistent due to perturbations.

- Tensor approach can be applied to square, under-determined and over-determined systems, and zero-dimensional and positive-dimensional systems.

- ALS algorithm is easy to implement and converges fast for generic tensors approximations. Additionally, ALS has also locally convergence properties. When the starting point is in the neighborhood of a global solution of (P5) then the iterates of ALS converge linearly to this solution [Uschmajew 2012].

- Although tensor approximation algorithms can not find all solutions of (P1), they can deal with systems up to 20 variables/equations in a reasonable time, contrary to algebraic geometry methods.

- Newton-based algorithms can not be applied to ill-conditioned systems, which is not a drawback to tensor approximations algorithms.

- Although the construction of $\boldsymbol{\mathcal{T}}'$ depends on the number of equations m, Problem (5) itself does not depend on m, which is an advantage compared to other methods that take the number of equations into account, such as those in [Schnabel 1984, Grigoriev 2005].

4.4 Extension to the complex field

The theory developped in Section 4.2 for real tensors can be extended to the following complex system:

$$\textbf{P6:} \begin{cases} \boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{b}_1^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{c}_1^{\mathsf{T}}\boldsymbol{x}^* + d_1 = 0 \\ \qquad\qquad \vdots \\ \boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}_m\boldsymbol{x} + \boldsymbol{b}_m^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{c}_m^{\mathsf{T}}\boldsymbol{x}^* + d_m = 0 \end{cases},$$

where $\boldsymbol{x} \in \mathbb{C}^n$, $\boldsymbol{A}_j \in \mathbb{C}^{n\times n}$, $\boldsymbol{b}_j \in \mathbb{C}^n$, $\boldsymbol{c}_j \in \mathbb{C}^n$, and $d_j \in \mathbb{C}$, for $j = 1, \ldots, m$.

Problem (P6) is not a general complex quadratic system since there is no terms of the form $\alpha_{ij}x_ix_j$ nor $\beta_{ij}x_i^*x_j^*$, for $\alpha_{ij}, \beta_{ij} \in \mathbb{C}$, $1 \leq i, j \leq n$. As a metter of fact, general quadratic complex systems cannot be formulated as rank-1 tensor approximation problems with the same approach detailed before, unless the systems should be separated in real and imaginary parts, in which case both the number of variables and equations are doubled. Besides, rewrite a complex system by taking separately the real and imaginary parts can be an unnecessary and annoying task, manly if the number of equations/unknowns is large enough. On the other hand, without doubling the number of variables, Problem (P6) can be reduced to a conjugate partially symmetric rank-1 tensor approximation problem, which can be solved by performing an algorithm called CPS (see Section 4.5).

The following lemma shows that each polynomial in (P6) is equivalent to two polynomial equations composed of Hermitian matrices.

Lemma 4.4.1 *Let* $\boldsymbol{y} = [\boldsymbol{x}^{\mathsf{T}}\ 1]^{\mathsf{T}} \in \mathbb{C}^{n+1}$. *For every* $j \in \{1, 2, \ldots, m\}$,

$$\boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}_j\boldsymbol{x} + \boldsymbol{b}_j^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{c}_j^{\mathsf{T}}\boldsymbol{x}^* + d_j = 0 \iff \begin{cases} \boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_j\boldsymbol{y} = 0 \\ \boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_{j+m}\boldsymbol{y} = 0, \end{cases}$$

where

$$\boldsymbol{Q}_j = \begin{bmatrix} \boldsymbol{A}_j + \boldsymbol{A}_j^{\mathsf{H}} & \boldsymbol{b}_j^* + \boldsymbol{c}_j \\ \boldsymbol{b}_j^{\mathsf{T}} + \boldsymbol{c}_j^{\mathsf{H}} & d_j + d_j^* \end{bmatrix} \text{ and } \boldsymbol{Q}_{j+m} = \frac{1}{i}\begin{bmatrix} \boldsymbol{A}_j - \boldsymbol{A}_j^{\mathsf{H}} & \boldsymbol{c}_j - \boldsymbol{b}_j^* \\ \boldsymbol{b}_j^{\mathsf{T}} - \boldsymbol{c}_j^{\mathsf{H}} & d_j - d_j^* \end{bmatrix},$$

with $i = \sqrt{-1}$.

Proof: The proof follows directly from the real and imaginary parts of each equation in (P6). Hence, we have

$$\begin{aligned} &\Re\{\boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}_j\boldsymbol{x} + \boldsymbol{b}_j^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{c}_j^{\mathsf{T}}\boldsymbol{x}^* + d_j\} = 0 \\ &\quad\iff \boldsymbol{x}^{\mathsf{H}}(\boldsymbol{A}_j + \boldsymbol{A}_j^{\mathsf{H}})\boldsymbol{x} + (\boldsymbol{b}_j^{\mathsf{T}} + \boldsymbol{c}_j^{\mathsf{H}})\boldsymbol{x} + (\boldsymbol{c}_j^{\mathsf{T}} + \boldsymbol{b}_j^{\mathsf{H}})\boldsymbol{x}^* + d_j + d_j^* = 0 \\ &\quad\iff \boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_j\boldsymbol{y} = 0, \end{aligned}$$

and

$$\begin{aligned}
&\Im\{\boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}_j\boldsymbol{x} + \boldsymbol{b}_j^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{c}_j^{\mathsf{T}}\boldsymbol{x}^* + d_j\} = 0 \\
&\quad\iff \frac{1}{i}\left(\boldsymbol{x}^{\mathsf{H}}(\boldsymbol{A}_j - \boldsymbol{A}_j^{\mathsf{H}})\boldsymbol{x} + (\boldsymbol{b}_j^{\mathsf{T}} - \boldsymbol{c}_j^{\mathsf{H}})\boldsymbol{x} + (\boldsymbol{c}_j^{\mathsf{T}} - \boldsymbol{b}_j^{\mathsf{H}})\boldsymbol{x}^* + d_j - d_j^*\right) = 0 \\
&\quad\iff \boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_{j+m}\boldsymbol{y} = 0.
\end{aligned}$$

□

We can define $\tilde{p}(\boldsymbol{y}) = \sum_{j=1}^{2m} |\boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_j\boldsymbol{y}|^2$, and write the following optimization problem

$$\textbf{P7:}\ \begin{cases} p_7^\star = \min\limits_{\boldsymbol{y}} \tilde{p}(\boldsymbol{y}) \\ \\ s.t.\ \|\boldsymbol{y}\| = 1. \end{cases}$$

(P6) and (P7) are connected problems as (P1) and (P2) are for the real case, which means that the solutions of System (P6) are minimizers of (P7). Yet, by defining $\tilde{\boldsymbol{M}}_Q = \sum_{j=1}^{2m} \boldsymbol{Q}_j^* \boxtimes \boldsymbol{Q}_j$, we can draw up the equivalent problem

$$\textbf{P8:}\ \begin{cases} p_8^\star = \max\limits_{\boldsymbol{y}} (\boldsymbol{y}^* \boxtimes \boldsymbol{y})^{\mathsf{H}}\tilde{\boldsymbol{M}}(\boldsymbol{y}^* \boxtimes \boldsymbol{y}) \\ \\ s.t.\ \|\boldsymbol{y}\| = 1, \end{cases}$$

where $\tilde{\boldsymbol{M}} \in \mathbb{C}^{(n+1)^2\times(n+1)^2}$ is a semidefinite positive matrix given by $\tilde{\boldsymbol{M}} = \lambda_{max}\boldsymbol{I} - \tilde{\boldsymbol{M}}_Q$, with λ_{max} the largest eigenvalue of $\tilde{\boldsymbol{M}}_Q$. Thus, the equivalence between (P2) and (P3) is extended to the complex case with (P7) and (P8).

Remark 3 *We could define $\tilde{p}(\boldsymbol{y}) = \sum_{j=1}^{2m}(\boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_j\boldsymbol{y})^2$ since Q_j are Hermitian matrices, which would lead to a cost function for (P8) given by $(\boldsymbol{y}\boxtimes\boldsymbol{y})^{\mathsf{H}}\tilde{\boldsymbol{M}}(\boldsymbol{y}\boxtimes\boldsymbol{y})$, with $\tilde{\boldsymbol{M}}_Q = \sum_{j=1}^{2m}\boldsymbol{Q}_j\boxtimes\boldsymbol{Q}_j$. However, as we will see in Section 4.5, it would imply to conceive a more complex algorithm to tackle complex tensors.*

The same reasoning used to construct the partially symmetric rank-1 approximation (P4) from (P3) can be applied here. We omit again the details because they are identical to those of the real case. Thus, (P8) is equivalent to the rank-1 tensor problem given by

$$\textbf{P9:}\ \begin{cases} p_9^\star = \min\limits_{\boldsymbol{y},\boldsymbol{w}} \|\boldsymbol{\mathcal{T}} - \boldsymbol{y}\otimes\boldsymbol{y}^*\otimes\boldsymbol{w}\|. \\ \\ s.t.\ \|\boldsymbol{y}\| = 1, \end{cases}$$

with $\boldsymbol{\mathcal{T}} \in \mathbb{C}^{n+1\times n+1\times\tilde{K}}$, and $\tilde{K} \leq \operatorname{rank}\{\tilde{\boldsymbol{M}}\}$. Problem (P9) is actually a *conjugated partially symmetric rank-1 approximation*.

Theorem 1 in [Friedland 2013] cannot be extended to conjugated partially symmetric complex tensors[2]. In other words, we cannot ensure that a best rank-1 approximation of a such tensor can be

[2]Even to partially symmetric complex tensors.

chosen conjugated partially symmetric. Hence, we cannot derive a best rank-1 approximation problem as (P5), so that we should directly tackle Problem (P9). For this, we propose an algorithm called CPS, based on the stationary equations of (P9). This algorithm is described in the next section.

4.5 Conjugated partially symmetric rank-1 approximation

In order to compute the best conjugated partially symmetric rank-1 approximation of a tensor in the complex field, we propose an alternating algorithm, based on the stationary equations of the Lagrangian of the matrix slice form of Problem (P9), which follows

$$\mathcal{L} = \mu \left(\|\boldsymbol{y}\|^2 - 1 \right) + \sum_{k=1}^{\tilde{K}} \|\boldsymbol{T}_k - w_k \boldsymbol{y}\boldsymbol{y}^{\mathsf{H}}\|^2.$$

The stationary equations are

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial \mu} = 0 \implies \|\boldsymbol{y}\|^2 = 1. \\ \dfrac{\partial \mathcal{L}}{\partial w_k^*} = 0 \implies w_k = \dfrac{\boldsymbol{y}^{\mathsf{H}} \boldsymbol{T}_k \boldsymbol{y}}{\|\boldsymbol{y}\|^4}, \ \forall k. \\ \dfrac{\partial \mathcal{L}}{\partial \boldsymbol{y}} = 0 \implies \sum_{k=1}^{\tilde{K}} \left(w_k^* \boldsymbol{T}_k + w_k \boldsymbol{T}_k^{\mathsf{H}} - 2|w_k|^2 \|\boldsymbol{y}\|^2 \boldsymbol{I} \right) \boldsymbol{y} = \mu^* \boldsymbol{y}. \end{cases} \tag{4.7}$$

By using the first stationary equation of (4.7), it follows that

$$w_k = \boldsymbol{y}^{\mathsf{H}} \boldsymbol{T}_k \boldsymbol{y} \ \text{ and } \ \sum_{k=1}^{\tilde{K}} \left(w_k^* \boldsymbol{T}_k + w_k \boldsymbol{T}_k^{\mathsf{H}} - 2|w_k|^2 \boldsymbol{I} \right) \boldsymbol{y} = \mu^* \boldsymbol{y}. \tag{4.8}$$

By multiplying both sides of the second equation of (4.8) by $\boldsymbol{y}^{\mathsf{H}}$, we obtain $\mu = 0$. Hence,

$$\sum_{k=1}^{\tilde{K}} \left(w_k^* \boldsymbol{T}_k + w_k \boldsymbol{T}_k^{\mathsf{H}} \right) \boldsymbol{y} = \gamma \boldsymbol{y}, \tag{4.9}$$

where $\gamma = 2 \sum_{k=1}^{\tilde{K}} |w_k|^2$.

Notice that the Lagrangian can be rewritten as $\mathcal{L} = \|\boldsymbol{\mathcal{T}}\|^2 - \gamma$, which reveals that to minimize $\mathcal{L}$, we need to maximize γ. The maximization of γ in equation (4.9) and the equations $w_k = \boldsymbol{y}^{\mathsf{H}} \boldsymbol{T}_k \boldsymbol{y}$, $1 \leq k \leq \tilde{K}$ can be solved alternately by the algorithm depicted in Alg.6. As a metter of fact, for some $\boldsymbol{y}$, we compute all w_k and substitute them in equation (4.9). Thus, the update of $\boldsymbol{y}$ is performed by taking the dominant eigenvector of $\sum_{k=1}^{\tilde{K}} \left(w_k^* \boldsymbol{T}_k + w_k \boldsymbol{T}_k^{\mathsf{H}} \right)$, since γ must be maximized.

Some remarks about the CPS algorithm:

input : $\boldsymbol{\mathcal{T}} \in \mathbb{C}^{n+1\times n+1\times \tilde{K}}$.
output: $\boldsymbol{y}, w_k,$ with $k \in \{1, \tilde{K}\}$.
Initialize $\boldsymbol{y}$;
repeat
  $\boldsymbol{V} \leftarrow \mathbf{0}$;
  for $k = 1$ *to* $\tilde{K}$ **do**
    $w_k \leftarrow \boldsymbol{y}^{\mathsf{H}}\boldsymbol{T}_k\boldsymbol{y}$;
    $\boldsymbol{V} \leftarrow \boldsymbol{V} + w_k^*\boldsymbol{T}_k + w_k\boldsymbol{T}_k^{\mathsf{H}}$;
  end
  Set $\boldsymbol{y}$ as the eigenvector of $\boldsymbol{V}$ whose eigenvalue is maximum;
until *some stopping criterion is satisfied*;

Algorithm 6: CPS algorithm

- As described in Remark 3, if the definition of $\tilde{p}(\boldsymbol{y})$ were $\tilde{p}(\boldsymbol{y}) = \sum_{j=1}^{2m}(\boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_j\boldsymbol{y})^2$, (P9) would be a partially symmetric problem. However, the Lagrangian would be

$$\mathcal{L} = \mu\left(\|\boldsymbol{y}\|^2 - 1\right) + \sum_{k=1}^{\tilde{K}} \|\boldsymbol{T}_k - w_k\boldsymbol{y}\boldsymbol{y}^{\mathsf{T}}\|^2,$$

  in which case the stationary equation (4.9) would not be an eigevalue problem on $\boldsymbol{y}$. The problem could still be adapted to an eigenvalue problem but with a more complex matrix;

- The complexity per iteration of CPS algorithm is given by $\mathcal{O}\{2(\tilde{K}+\rho)(n+1)^2\}$, where ρ is the number of steps[3] used in Lanczo's algorithm to compute the dominant eigenvector. For large n, $\tilde{K}$ is also large, so that ρ can be dropped of the complexity, since it is generally a very small value;

- The CPS algorithm can also be used to tackle the partially symmetric rank-1 approximation problem (P4). In this case, the equations on w_k and γ becomes $w_k = \boldsymbol{y}^{\mathsf{T}}\boldsymbol{T}_k\boldsymbol{y}$ and $\sum_{k=1}^{K} w_k(\boldsymbol{T}_k + \boldsymbol{T}_k^{\mathsf{T}})\boldsymbol{y} = \gamma\boldsymbol{y}$.

4.6 Examples on real and complex fields

In this section, we present some examples of solving multivariate quadratic systems. In the real case, we use the standard ALS algorithm whereas for complex systems the algorithm used is the proposed CPS. For both algorithms the stopping criterion is satisfied when either the maximum number of iterations ($itr_{max} = 10000$) or the threshold $|\|\boldsymbol{\mathcal{T}} - \hat{\boldsymbol{\mathcal{T}}}_{[itr]}\| - \|\boldsymbol{\mathcal{T}} - \hat{\boldsymbol{\mathcal{T}}}_{[itr-1]}\|| \leq 10^{-6}$ for an estimate $\hat{\boldsymbol{\mathcal{T}}}$ at iteration itr, is attained.

Since CPS is also an initialization-dependent algorithm, we test four initialization algorithms: SeROAP, THOSVD and ST-HOSVD (described in Chapter 2), and a conjugated random initialization method. The latter consists of the selection of the tensor that minimizes the approximation error from a sample of rank-1 tensors with conjugated factors, generated according to a uniform probability

[3] In chapters 1 and 2, we have used the variable k instead. Here k is already used as the index of the third mode of $\boldsymbol{\mathcal{T}}$.

distribution with entries in $[-1, 1]$. Let L be the size of this sample and $\boldsymbol{y}^{(l)}, \boldsymbol{w}^{(l)}, l \in \{1, 2, \ldots, L\}$, be the factors of the l^{th}-generated tensor. The factor $\boldsymbol{y}$ is initialized in CPS algorithm as follows

$$\boldsymbol{y} = \arg\min_{\boldsymbol{y}^{(l)}} \|\boldsymbol{\mathcal{T}} - \boldsymbol{y}^{(l)} \otimes \boldsymbol{y}^{(l)*} \otimes \boldsymbol{w}^{(l)}\|.$$

The initializations can also be applied to the ALS algorithm, so that the conjugated random initialization becomes a symmetric random initialization. The experiments have performed with Matlab© in an MAC OSX 10.8.5 with processor 3.2 GHz Intel core i5, and memory 8G 1600MHz DDR3.

Example 1: Let the overdetermined real system with 5 unknowns and 7 equations

$$\begin{cases} x_1^2 + x_2x_3 + x_5 = 3 \\ x_2^2 + x_2x_5 - 2x_3x_4 + x_5^2 = 5 \\ x_1x_3 + x_1x_5 - x_4^2 + x_2 - x_4 = 0.25 \\ x_1x_2 + x_3^2 + x_3x_5 - 3x_2 = -1 \\ x_1x_4 - 3x_1x_5 + 2x_2^2 + x_1 - x_4 = -5 \\ x_1^2 + x_3^2 + x_4^2 - x_5^2 + x_4x_5 = -0.75 \\ x_2^2 + x_1 + x_3 = 0 \end{cases}, \tag{4.10}$$

where $\boldsymbol{x} = [1\ 0\ -1\ 0.5\ 2]^\mathsf{T}$ is the real solution. The tensor $\boldsymbol{\mathcal{T}}'$ constructed from the system has dimensions $6 \times 6 \times 21$. The slices k' such that $\|\boldsymbol{T}_{k'}\| \leq 10^{-6}$, for $k' \leq 36$ were ignored.

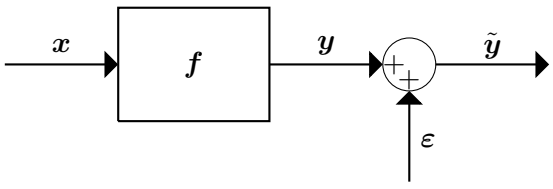
Table below shows the error of the estimation $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$, and the time of execution to compute the solution of (4.10), using ALS for four different initialization methods. The time to initialize the algorithm is not considered in the table.

Initialization	$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$	time ALS (seconds)
Random (L = 50000)	3.1030e-4	3.5652
THOSVD	3.5023e-4	2.7237
ST-HOSVD	3.2092e-4	2.8074
SeROAP	3.2088e-4	2.7705

Table 4.1: Error estimation for different initializations using ALS algorithm for a $(5, 7)$ system.

For all initializations, ALS delivers the approximative solution $\hat{\boldsymbol{x}} = [1\ 0\ -1.0003\ 0.4998\ 2]^\mathsf{T}$.

Example 2: For the same overdetermined real system, consider the following diagram representation:



where $\boldsymbol{f} = [f_1\ f_2\ \ldots\ f_5]^\mathsf{T}$, and $\boldsymbol{y} = [y_1\ y_2\ \ldots\ y_5]^\mathsf{T}$, such that $f_i(\boldsymbol{x}) = y_i$ is the i-th equation in (4.10). We estimate $\boldsymbol{x}$ for a given noisy output $\tilde{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is a perturbation vector

distributed according to a gaussian measure with variance σ^2. We use SeROAP to initialize ALS for all simulations.

Table below shows the average estimation $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$ for a sample of 1000 realizations with three different values of σ. We notice that the average estimation is approximately equal to σ, which evinces that our rank-1 tensor approach to solve the system is robust under perturbations.

σ	Average $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$	Average time ALS (seconds)
0.001	0.0011	3.0819
0.01	0.011	3.4510
0.1	0.1102	3.9467

Table 4.2: Average error estimation using ALS algorithm for a $(5, 7)$ noisy system.

Example 3: Let the under-determined real system with 6 unknowns and 3 equations

$$\begin{cases} 3x_1^2 + x_1x_2 - 3x_2^2 - x_2x_6 = 1 \\ -x_3^2 + 2x_4x_5 + x_1 + x_2 = 3 \\ x_1^2 + 3x_4^2 + 2x_5^2 + x_6^2 - 2x_2 = 4 \end{cases}, \tag{4.11}$$

Table 4.3 shows that one solution is extracted for ALS for all initializations. Indeed, $\max_j |\hat{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{Q}_j\hat{\boldsymbol{x}}|$ $\leq 1.027{\cdot}10^{-4}$, $1 \leq j \leq 2m$, for all initializations. The solution $\hat{\boldsymbol{x}} = [1.1843\ 1.8156\ 0\ 0\ 0\ -2.4957]^{\mathsf{T}}$ was obtained with the finite algorithms SeROAP, ST-HOSVD, and THOSVD. While with the random initialization we have extracted the solution $\hat{\boldsymbol{x}} = [1.6030\ 1.5943\ -1.1678\ 0.9735\ 0.5991\ 1.0286]^{\mathsf{T}}$.

Initialization	$\max_j \|\hat{\boldsymbol{x}}^{\mathsf{T}}\boldsymbol{Q}_j\hat{\boldsymbol{x}}\|$	Time ALS (seconds)
Random (L = 50000)	1.8113e-5	0.2187
THOSVD	8.8285e-5	0.8268
ST-HOSVD	9.1748e-5	0.8159
SeROAP	1.0270e-4	0.8632

Table 4.3: Error estimation for different initializations using ALS algorithm for a $(6, 3)$ system.

Example 4: Let the overdetermined complex system with 3 unknowns and 4 equations below

$$\begin{cases} 2|x_1|^2 + x_1^*x_2 + ix_3^* = 1 + i \\ -2x_1x_3^* + x_1^*x_3 + |x_2|^2 - x_2 = 1 - 2i \\ -i|x_3|^2 + x_2^*x_3 - (1+i)x_1 + x_2^* = 1 \\ -|x_1|^2 + 3|x_3|^2 + 3ix_2^* = -1 \end{cases}, \tag{4.12}$$

where $\boldsymbol{x} = [i\ \ -i\ 1]^{\mathsf{T}}$ is the complex solution. As mentioned in Section 4.4, one way to solve this system is to substitute each unknown by its complex form $a + bj$ and solve a real system with twice as many equations. However, this transformation introduces additional calculations, thereby we go along to solve the problem directly using the rank-1 tensor approach.

The tensor $\mathcal{T}$ constructed from the system has dimensions $4 \times 4 \times 16$. As in the real case, some slices of $\mathcal{T}$ were not taken into account because their norms are negligible. The estimation and time performance using the CPS algorithm are presented in table below. For all initializations, the algorithm extracts the solution $[i \ -i \ 1]^{\mathsf{T}}$.

Initialization	$\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$	Time CPS (seconds)
Random (L = 50000)	9.5133e-6	1.3620
THOSVD	9.5666e-6	1.8400
ST-HOSVD	9.9688e-6	1.8066
SeROAP	1.2052e-5	1.6464

Table 4.4: Error estimation for different initializations using CPS algorithm for complex systems.

4.7 Performance evaluation of rank-1 tensor approach

In this section, we present some numerical experiments at which a solution of generic quadratic systems are extracted using the standard ALS and CPS algorithms for real and complex rank-1 tensor approximations, respectively. For the real case, a total of 500 systems for each scenario (n, m) was generated with the entries of $\boldsymbol{A}_j, \boldsymbol{b}_j, c_j, \ 1 \leq j \leq m$, distributed according to a uniform measure in $[-1, 1]$. For complex systems, the entries of the real and imaginary parts of $\boldsymbol{A}_j, \boldsymbol{b}_j, \boldsymbol{c}_j, d_j, \ 1 \leq j \leq 2m$, are also uniformly distributed in $[-1, 1]$. We have employed the symmetric (or conjugated symmetric) random initialization.

Table 4.5 summarizes the percentage p of real systems in which a solution was successfully obtained, and the average computational time for initialization, construction of $\mathcal{T}'$, and the rank-1 approximation itself. We assume that a solution satisfies a system if

$$\max(|\boldsymbol{y}^{\mathsf{T}}\boldsymbol{Q}_1\boldsymbol{y}|, |\boldsymbol{y}^{\mathsf{T}}\boldsymbol{Q}_2\boldsymbol{y}|, \ldots, |\boldsymbol{y}^{\mathsf{T}}\boldsymbol{Q}_m\boldsymbol{y}|) \leq 0.001.$$

All time measurements are denoted in seconds.

The results confirms that a solution of an MQ system can be extracted in a few seconds with ALS algorithm. Notice that the computational time of the ALS algorithm is larger for systems having variables as many as equations. For instance, compare scenarios $(4, 4)$ and $(6, 6)$ with scenarios $(5, 4)$ and $(7, 9)$, respectively.

Scenario (n, m)	Initialization Time	Time for constructing $\mathcal{T}'$	Time ALS	Total Time	p (%)
$(2, 3)$	1.2707	8.5767e-04	0.0852	1.3567	100
$(4, 4)$	1.5190	0.0014	2.0137	3.5340	100
$(5, 4)$	1.5288	0.0024	0.4613	1.9925	100
$(6, 6)$	1.5800	0.0035	3.5453	5.1288	100
$(7, 9)$	1.6444	0.0051	0.9939	2.6433	98.6
$(10, 8)$	2.7514	0.0133	2.0391	4.8037	100
$(10, 10)$	2.7705	0.0133	8.8520	11.6358	100
$(20, 20)$	11.8718	0.1993	51.4871	63.5582	100

Table 4.5: Performance of ALS algorithm to extract one solution of generic quadratic systems.

Table 4.6 shows the performance of CPS for some of the aforementioned scenarios but in the complex field. We use the same metrics to evaluate our algorithm, except the condition in which a solution satisfies a system that now is given by $\max(|\boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_1\boldsymbol{y}|, |\boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_2\boldsymbol{y}|, \ldots, |\boldsymbol{y}^{\mathsf{H}}\boldsymbol{Q}_{2m}\boldsymbol{y}|) \leq 0.002$.

Scenario (n, m)	Initialization Time	Time for constructing $\mathcal{T}'$	Time CPS	Total Time	p (%)
$(2, 3)$	7.2546	9.9674e-04	1.0424	8.2980	96
$(4, 4)$	8.1409	0.0020	30.9283	39.0712	70.2
$(5, 4)$	8.7274	0.0030	9.0716	17.8020	97.2
$(10, 8)$	27.2700	0.0217	36.4495	63.7413	79

Table 4.6: Performance of CPS algorithm to extract one solution of generic quadratic systems.

The performance of CPS algorithm to compute a conjugated partially symmetric rank-1 approximation and thus extract one solution of System (P6) is not good as in the real case. We eliminate the scenarios where $p < 70\%$ so that only four scenarios are presented. Moreover, the initialization and CPS times are large compared to the real scenarios. As before, the square system also presents larger computational time among all scenarios.

4.8 Chapter Summary and Directions

We conclude this chapter by presenting the main points observed in the results and also we given some directions for future work.

- *Reduction of multivariate quadratic systems to best rank-1 three-way tensor approximation problems.* We showed that a general real MQ system can be reduced to a best rank-1 approximation of a three-way real tensor. The proof was delineated along four propositions in Section 4.2. This approach arises as a new mathematical tool to tackle MQ systems when one solution is suitable. Some discussions about the rank-1 tensor approach were outlined in Section 4.3. Therein, we saw that the rank-1 approximation itself does not depend on the number of equations, which means that the dimensions of a tensor $\boldsymbol{\mathcal{T}}$, constructed from a system of m equations, is independent of m. We also saw that there is no constraints of using our approach, contrary to some other methods, such as the Newton-based one in [Dennis Jr 1996], where the system must be square and nonsingular, or [Verschelde 1999], where the system must be squared before applying the homotopy continuation. On the other hand, the main limitation of the rank-1 tensor approach is the nonequivalence between problems (P1) and (P5). Although ALS extracts one solution of an MQ system in general, we do not have yet a complete answer for the following questions: (i) which cases $y_{n+1} = 0$ in (P2)? (ii) What is the set of tensors at which $\mathcal{S}_{P4} = \mathcal{S}_{P5}$? For (i), if we map all non trivial solutions of (P1) when $c_j = 0,\ 1 \leq j \leq m$, then all solutions of (P2) such that $y_{n+1} = 0$ are also mapped. We intend to study and perform this mapping in a future work. For (ii), we claim that the set of partially symmetric tensors whose rank-1 approximations are also partiality symmetric has volume measure equal to 1, but no proof was presented here. Nevertheless, the iterative ALS algorithm applied to a partially

symmetric tensor delivers a partially symmetric rank-1 approximation in general. Thus, from an algorithmic point of view, the nonequivalence between problems (P1) and (P5) does not impose any issue.

- *Extension to complex quadratic systems.* we extended our rank-1 tensor approach to complex MQ systems with no monomials of the form $x_i x_j$ or $x_i^* x_j^*$, for $1 \leq i, j \leq 2m$. Indeed, we saw that such complex systems can be reduced to conjugated partially symmetric rank-1 three-way approximation problems. As discussed in Section 4.4, any complex MQ system is equivalent to a real MQ system with twice the number of variables and equations. However, with our rank-1 tensor approach, we kept the same number of variables as in the real case.

- *Proposition of the CPS algorithm.* In order to deal with the conjugated partially symmetric rank-1 approximation problem, we proposed an alternating algorithm called CPS, based on the stationary equations of (P9), which can also be applied to real tensors.

- *Performance of the rank-1 tensor approach.* Numerical experiments showed that ALS algorithm extract a solution even for real systems of 20 variables/unknowns in a reasonable time, which is not possible with standard algebraic geometry methods. In the complex case, we saw that CPS algorithm is also capable of obtaining conjugated partially symmetric rank-1 approximations but the performance is poorer than that obtained for real systems. We presented some examples of real with/without perturbations, an example of complex system, and also a set of Monte Carlo experiments for different sizes of systems to evince the efficiency of the rank-1 tensor approximation approach.

CHAPTER 5

Alternating projections on orthogonal CP tensor decomposition

This chapter concerns the exact decomposition of tensors with one semi-unitary factor matrix. Firstly, we present a brief discussion on the alternating projection method and the orthogonal tensor decomposition. We emphasize some applications and theoretical results in the literature.

Next, the description of our algorithm, called CAPD, combining the alternating projection method and the deflation procedure is presented in details. We develop the algorithm based on the set (manifold) of unit rank-1 matrices and a linear subspace. In the sequel, we perform a convergence study of the CAPD algorithm, whose main features are the extraction of the right factors, and the convergence under the transversality and non-tangential concepts. In this part, we resort to the main theorems of [Lewis 2008, Andersson 2013] pointing out the convergence to the intersection of the manifolds. These results are contextualized to the column-wise tensor decomposition, so that the convergence to the exact decomposition is ensured.

Finally, we draw some numerical experiments evincing the efficiency of our proposed algorithm. We finish this chapter presenting some conclusions and putting out future works.

Main contributions of this chapter:

- *CAPD algorithm.* We propose an algorithm, called CAPD, that combines the alternating projection method and the deflation of rank-1 tensors to solve the exact decomposition of a three-way tensor with one semi-unitary factor matrix. The alternating projection part consists of iterative projections onto the manifold of unitary rank-1 matrices and a linear subspace, aiming the computation of one column of each factor matrix. Thus, we can construct a rank-1 component of the tensor and deflated this component from the original tensor, so that the rank is reduced by 1. The process is repeated until all components can be extracted.

- *Extraction of the right rank-1 component.* We proof that a right rank-1 component is extracted by our proposed algorithm if the alternating projection part converges to a point in the intersection of the manifolds. This is shown in Theorem 5.4.3.

- *Convergence under the transversality concept.* Under some conditions, we prove that if points in the intersection of the manifolds involved in the column-wise orthogonal tensor decomposition are transversal, and the starting point of CAPD algorithm is close to that intersection, then the linear convergence of our algorithm is ensured. This results is presented in Proposition 5.4.7. However, the transversality concept applied to our problem imposes some limitations on

the dimensions of the manifolds. Actually, apart from a special case, only a few components of the tensor can be extracted.

- *Convergence under the non-tangential concept.* We prove a stronger result based on the non-tangential concept. For the tensor decomposition problem with an orthogonal factor matrix, we show that CAPD algorithm converges to a point in the intersection of the manifolds, when the starting point is close enough to this intersection, and the manifolds are non-tangential at any point in the intersection. This approach mitigates the limitation of the transversality concept since now there is no restriction on the dimensions of the manifolds. This is depicted in Proposition 5.4.9.

- *Performance of CAPD algorithm.* Finally, we show by simulations that the distance of iterates of the CAPD algorithm converges log-linearly. We also show the drawbacks of estimating the factor matrices all-at-once using the alternating projection method. Yet, the results evince that CAPD is suitable for solving the aforementioned tensor problem.

Contents

5.1 An introduction on alternating projections

The method of alternating projections consists of projecting iteratively a point onto manifolds, one after the other, in order to find a point in their intersection. To describe the method, let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n$ be n manifolds in some space $\mathcal{E}$. The alternating projection is given as follows

$$\boldsymbol{x}^{(k+1)} = \pi_{\mathcal{M}_n} \cdots \pi_{\mathcal{M}_2} \pi_{\mathcal{M}_1}(\boldsymbol{x}^{(k)}),\ \forall k = 0, 1, \ldots \tag{5.1}$$

where $\pi_{\mathcal{M}_i},\ 1 \leq i \leq n$ is the orthogonal projection operator onto $\mathcal{M}_i$, and k is the iteration.

This method was originally proposed by von Neumann in 1930's for two manifolds, and he also proved that the sequence of alternating projections of two linear manifolds converges to a point in the intersection [Von Neumann 1950].

Method 5.1 has also been much emphasized in the literature when the problem has two convex manifolds. We highlight, for instance, the articles [Combettes 1990, Higham 2002, Gubin 1967, Breg 1965] for some discussion and application of the alternating convex projection method.

The extension of alternating projections to nonconvex manifolds plays an important role in several applications. We cite, for instance, those in control design [Grigoriadis 1999] and in image processing [Bauschke 2002]. For $n = 2$, some important theoretical results on convergence of the alternating nonconvex projection method are highlighted in the following.

In [Zangwill 1967], it is shown that if the sequence $\{\boldsymbol{x}^{(k)}\}_{k=0}^{\infty}$ is bounded and the distance to $\mathcal{M}_1 \cap \mathcal{M}_2$ is strictly decreasing, then there exists a convergent subsequence to a point in $\mathcal{M}_1 \cap \mathcal{M}_2$. This results is known as the Zangwill's global conergence theorem. However, it is a weak result since it does not provide any information about the convergence of the complete sequence.

The authors in [Lewis 2008] introduce the concept of transversal manifolds and show that a linear convergence of the alternating projection method can be ensured if the manifolds are transversal at some point in $\mathcal{M}_1 \cap \mathcal{M}_2$ and the starting point $\boldsymbol{x}^{(0)}$ is close enough to $\mathcal{M}_1 \cap \mathcal{M}_2$. However, this approach presents a restriction on the dimensions of the manifolds. In order to drop the limitation of the traversal concept, the authors in [Andersson 2013] proposed a new concept called *non-tangencial intersection point*, which generalizes that of transversality. Both the transversality and non-tangential concepts will be studied in the context of orthogonal tensor decomposition. Some results of these two last publications are presented here and contextualized to our specific problem.

5.2 Column-wise orthogonal tensor decomposition

A tensor is orthogonal if its rank-1 decomposable components are orthogonal to one another, i.e., $\langle \boldsymbol{\mathcal{X}}_i, \boldsymbol{\mathcal{X}}_j \rangle = 0,\ \forall i \neq j$, for the rank-$R$ tensor $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{X}}_1 + \boldsymbol{\mathcal{X}}_2 + \cdots + \boldsymbol{\mathcal{X}}_R$, where $\boldsymbol{\mathcal{X}}_i$ are rank-1 tensors. The notion of orthogonality between a couple of decomposable tensors depends on the level of coupling between their factors. To be clearer, consider two three-way decomposable tensors $\boldsymbol{\mathcal{X}}_1 = \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \otimes \boldsymbol{c}_1$ and $\boldsymbol{\mathcal{X}}_2 = \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \otimes \boldsymbol{c}_2$. The tensors $\boldsymbol{\mathcal{X}}_1$ and $\boldsymbol{\mathcal{X}}_2$ are completely orthogonal if $\boldsymbol{a}_1 \perp \boldsymbol{a}_2, \boldsymbol{b}_1 \perp \boldsymbol{b}_2$ and $\boldsymbol{c}_1 \perp \boldsymbol{c}_2$. Actually, all factors do not need to be orthogonal to one another to ensure the orthogonality of the tensor. For instance, if only $\boldsymbol{a}_1 \perp \boldsymbol{a}_2$, then $\langle \boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2 \rangle = 0$. Indeed,

$\langle \boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2 \rangle = \langle \boldsymbol{a}_1, \boldsymbol{a}_2 \rangle \langle \boldsymbol{b}_1, \boldsymbol{b}_2 \rangle \langle \boldsymbol{c}_1, \boldsymbol{c}_2 \rangle$. Herein, for a rank-$R$ tensor, we assume that the factor matrix $\boldsymbol{A}$ of a tensor $\boldsymbol{\mathcal{T}}$ is semi-unitary[1], which implies that $\langle \boldsymbol{a}_i, \boldsymbol{a}_j \rangle = \delta(i-j)$, ensuring the orthogonality of $\boldsymbol{\mathcal{T}}$. We call *column-wise orthogonal tensor decomposition* the decomposition of a tensor whose at least one of the factors matrices is semi-unitary. Notice that if all factor matrices are semi-unitary then the tensor is of unit norm and completely orthogonal. For other notions of orthogonality, see the article [Kolda 2001].

The column-wise orthogonal tensor decomposition plays an important role in several applications, such as blind source separation [De Lathauwer 2007], array processing[Miron 2008],and wireless systems [Sørensen 2010]. Some important results are drawn in the literature. In [Wang 2015], the authors show that the best orthogonal low rank approximation of a tensor, with one semi-unitary factor matrix, always exists. Additionally, for almost all tensors, they also show that the ALS algorithm converges globally. In [Sørensen 2012], the authors discuss the uniqueness and low-rank approximation properties of a CP decomposition with a semi-unitary matrix factor. In the following, we apply the results of alternating projection methods to solve this kind of decomposition.

5.3 Combined deflation and alternating projection algorithm

Let $\boldsymbol{A} \in \mathbb{K}^{I \times R}$, $\boldsymbol{B} \in \mathbb{K}^{J \times R}$ and $\boldsymbol{C} \in \mathbb{K}^{K \times R}$ be the factor matrices of a tensor $\boldsymbol{\mathcal{T}}$ with entries in some field $\mathbb{K}$. We assume that $\mathrm{rank}(\boldsymbol{\mathcal{T}}) = R$ is known, and that one of the factor matrices is column-wise orthogonal, say $\boldsymbol{A}$. This implies $R \leq I$. Actually, without loss of generality, $\boldsymbol{A}$ can be viewed as a semi-unitary matrix due to scalar elementary indeterminacies (unitary if $R = I$).

Let $\boldsymbol{T} = \boldsymbol{A}\,(\boldsymbol{C} \odot \boldsymbol{B})^{\mathsf{T}}$ be the mode-1 unfolding of tensor $\boldsymbol{\mathcal{T}}$. Since $\boldsymbol{A}$ is semi-unitary, $\boldsymbol{A}^{\mathsf{H}}\boldsymbol{A} = \boldsymbol{I}$. Therefore, after elementary operations

$$\boldsymbol{T}^{\mathsf{H}}\boldsymbol{A} = \boldsymbol{C}^* \odot \boldsymbol{B}^* \tag{5.2}$$

For a specific column of $\boldsymbol{A}$, namely $\boldsymbol{a}_r$, we derive from equation (5.2) that

$$\boldsymbol{T}^{\mathsf{H}}\boldsymbol{a}_r = \boldsymbol{c}_r^* \boxtimes \boldsymbol{b}_r^*, \tag{5.3}$$

where $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$ are the r-th column of the matrices $\boldsymbol{B}$ and $\boldsymbol{C}$, respectively.

We propose a combined alternating projection and deflation algorithm, called CAPD, that recovers all factor matrices of a given orthogonal tensor $\boldsymbol{\mathcal{T}}$ with a semi-unitary factor. The idea is to solve first the equation 5.3 with an alternating projection algorithm in order to estimate the factors $\boldsymbol{a}_r, \boldsymbol{b}_r$ and $\boldsymbol{c}_r$, and to perform the deflation of the rank-1 component $\boldsymbol{a}_r \otimes \boldsymbol{b}_r \otimes \boldsymbol{c}_r$ from $\boldsymbol{\mathcal{T}}$, obtaining thus a rank-$(R-1)$ tensor $\boldsymbol{\mathcal{T}} - \boldsymbol{a}_r \otimes \boldsymbol{b}_r \otimes \boldsymbol{c}_r$. The process is repeated[2] $R-2$ times until all rank-1 components are estimated.

Let $\boldsymbol{T}_{[r]}$ be the mode-1 unfolding of a rank-$(R-r+1)$ tensor in $\mathbb{K}^{I \times J \times K}$, with $1 \leq r \leq R$. We define the following linear subspace

[1] The upcoming reasoning is identical if $\boldsymbol{B}$ or $\boldsymbol{C}$ were semi-unitary.

[2] After the last deflation, the algorithm yields a rank-1 tensor, whose factors are easily obtained by computing an economic SVD.

$$\mathcal{M}_r = \{\boldsymbol{x} \in \mathbb{K}^{JK} : \left(\boldsymbol{T}_{[r]}^{\mathsf{H}} \boldsymbol{T}_{[r]}^{\mathsf{H}+} - \boldsymbol{I}\right) \boldsymbol{x} = \boldsymbol{0}\}, \tag{5.4}$$

and the manifold

$$\mathcal{N} = \{\boldsymbol{x} \in \mathbb{K}^{JK} : \boldsymbol{x} = \boldsymbol{c} \boxtimes \boldsymbol{b}, \|\boldsymbol{x}\| = 1, \forall \boldsymbol{b} \in \mathbb{K}^{J} \setminus \{\boldsymbol{0}\} \text{ and } \forall \boldsymbol{c} \in \mathbb{K}^{K} \setminus \{\boldsymbol{0}\}\}, \tag{5.5}$$

which are the column space of $\boldsymbol{T}_{[r]}^{\mathsf{H}} \boldsymbol{T}_{[r]}^{\mathsf{H}+}$ and the subset of unit Kronecker vectors in $\mathbb{K}^{JK}$, respectively. By applying the Unvec operator on $\boldsymbol{x}$, $\mathcal{N}$ can be actually viewed as the set of $J \times K$ unit-norm rank-1 matrices. In the proposed CAPD algorithm, the factors $\boldsymbol{a}_r, \boldsymbol{b}_r$ and $\boldsymbol{c}_r$ are computed by using the alternating projection onto these two manifolds. We set initially $\boldsymbol{T}_{[1]} = \boldsymbol{T}$. CAPD is described in Algorithm (7). The upper indices denote the k-th iteration of the algorithm, and $\pi_{\mathcal{M}_r}$ and $\pi_{\mathcal{N}}$ are the projection operators onto $\mathcal{M}_r$ and $\mathcal{N}$, respectively. Notice that $\pi_{\mathcal{M}_r} = \boldsymbol{T}_{[r]}^{\mathsf{H}} \boldsymbol{T}_{[r]}^{\mathsf{H}+}$.

input : $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I \times J \times K}$: input data
output: $\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{B}}$ and $\widehat{\boldsymbol{C}}$: factor matrices
$\boldsymbol{T} \leftarrow$ mode-1 unfolding of $\boldsymbol{\mathcal{T}}$;
$\boldsymbol{T}_{[1]} \leftarrow \boldsymbol{T}$;
Define $\mathcal{N}$;
for $r = 1$ *to* $R - 1$ **do**
  Define the linear subspace $\mathcal{M}_r$ from $\boldsymbol{T}_{[r]}$;
  Initialize $\boldsymbol{b}_r^{(0)}$ and $\boldsymbol{c}_r^{(0)}$;
  $\boldsymbol{x}_r^{(0)} \leftarrow \pi_{\mathcal{M}_r}(\boldsymbol{c}_r^{(0)} \boxtimes \boldsymbol{b}_r^{(0)})$;
  $k \leftarrow 0$;
  repeat
    $k \leftarrow k + 1$;
    $\boldsymbol{x}_r^{(k)} \leftarrow \pi_{\mathcal{M}_r} \pi_{\mathcal{N}}(\boldsymbol{x}_r^{(k-1)})$;
  until *some stopping criteria are satisfied*;
  $\widehat{\boldsymbol{a}}_r \leftarrow \boldsymbol{T}_{[r]}^{\mathsf{H}+} \boldsymbol{x}_r^{(k)}$;
  $\widehat{\boldsymbol{b}}_r \leftarrow$ conjugate of the left singular vector of $\mathrm{Unvec}(\boldsymbol{x}_r^{(k)})$;
  $\widehat{\boldsymbol{c}}_r \leftarrow$ right singular vector of $\mathrm{Unvec}(\boldsymbol{x}_r^{(k)})$;
  $\widehat{\boldsymbol{A}}(:, r) \leftarrow \widehat{\boldsymbol{a}}_r$; $\widehat{\boldsymbol{B}}(:, r) \leftarrow \widehat{\boldsymbol{b}}_r$; $\widehat{\boldsymbol{C}}(:, r) \leftarrow \widehat{\boldsymbol{c}}_r$;
  $\boldsymbol{T}_{[r+1]} \leftarrow \boldsymbol{T}_{[r]} - \widehat{\boldsymbol{a}}_r (\widehat{\boldsymbol{c}}_r \boxtimes \widehat{\boldsymbol{b}}_r)^{\mathsf{T}}$;
end
$\boldsymbol{T}_{[R]}$ is the mode-1 unfolding of the last rank-1 component of $\boldsymbol{\mathcal{T}}$.

Algorithm 7: CAPD algorithm

The algorithm works as follows. For every $r \in \{1, 2, \ldots, R-1\}$, the vectors $\boldsymbol{b}_r^{(0)}$ and $\boldsymbol{c}_r^{(0)}$ are randomly initialized so that $\boldsymbol{c}_r^{(0)} \boxtimes \boldsymbol{b}_r^{(0)} \notin \ker(\boldsymbol{T}_{[r]})$, otherwise $\boldsymbol{x}_r^{(0)} = \boldsymbol{0}$ and the algorithm would not extract the r-th rank-1 component. The vector $\boldsymbol{x}_r^{(0)}$ is introduced into a *repeat* loop that consists of the alternating projection phase. If $\boldsymbol{x}_r^{(k)}$ converges to a point in $\mathcal{M}_r \cap \mathcal{N}$, then the estimated factor $\widehat{\boldsymbol{a}}_r$ is computed by solving in the least square sense the equation (5.3) for a given $\boldsymbol{x}_r^{(k)} = \boldsymbol{c}_r^{(k)} \boxtimes \boldsymbol{b}_r^{(k)}$.

Clearly, the factors $\widehat{\boldsymbol{b}}_r$ and $\widehat{\boldsymbol{c}}_r$ are directly obtained from the singular vectors of $\mathrm{Unvec}(\boldsymbol{x}_r^{(k)})$ as shown in Algorithm (7).

After the estimation of $\widehat{\boldsymbol{a}}_r, \widehat{\boldsymbol{b}}_r$, and $\widehat{\boldsymbol{c}}_r$, we deflate the original tensor by the rank-1 component $\widehat{\boldsymbol{a}}_r(\widehat{\boldsymbol{c}}_r \boxtimes \widehat{\boldsymbol{b}}_r)^{\mathsf{T}}$, and perform again $R-2$ times the alternating projection procedure in order to find the other components of the factor matrices $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$.

Remark 4 *Neither the convergence of the CAPD algorithm to a point $\boldsymbol{x}_r^\star \in \mathcal{M}_r \cap \mathcal{N}$ nor the estimation of the right factors $\boldsymbol{a}_r, \boldsymbol{b}_r$ and $\boldsymbol{c}_r$ can be ensured. However, theoretical results in the next section guarantee local convergence and the extraction of the suitable factors.*

Remark 5 *Instead of estimating one-by-one the rank-1 components, we could try to estimate all components all-at-once from equation (5.2). However, the orthogonality of the estimated factor $\widehat{\boldsymbol{A}}$ would not be ensured anymore unless a constraint is imposed. Actually, all columns of the matrix factors could be collinear, as we will show in Section 5.5. Moreover, the estimation of all factors all-at-once requires more computations.*

5.4 Convergence study on CAPD algorithm

This section deals with important results on the convergence of CAPD algorithm. We start with the proof of some essential lemmas, which will be important to demonstrate that the extraction of one rank-1 component of the tensor is ensured if CAPD method converges to some point in $\mathcal{M}_r \cap \mathcal{N}$ for every $r \in \{1, 2, \ldots, R-1\}$. Finally, based on the ideas within [Lewis 2008], we show that CAPD algorithm has locally linear convergence properties.

Lemma 5.4.1 *Let $\boldsymbol{c}_r^{(k)} \boxtimes \boldsymbol{b}_r^{(k)} = \pi_{\mathcal{N}}(\boldsymbol{x}_r^{(k-1)})$ be the unit rank-1 approximation of vector $\boldsymbol{x}_r^{(k-1)}$ at k-th iteration of CAPD algorithm. If $\boldsymbol{c}_r^{(0)} \boxtimes \boldsymbol{b}_r^{(0)} \notin \ker(\boldsymbol{T}_{[r]})$ then for every $k \geq 1$, $\boldsymbol{c}_r^{(k)} \boxtimes \boldsymbol{b}_r^{(k)} \notin \ker(\boldsymbol{T}_{[r]})$ either.*

Proof: $\boldsymbol{c}_r^{(0)} \boxtimes \boldsymbol{b}_r^{(0)} \notin \ker(\boldsymbol{T}_{[r]}) = \ker(\boldsymbol{T}_{[r]}^{\mathsf{H}+}) \implies \boldsymbol{x}_r^{(0)} \in \mathrm{Ran}(\boldsymbol{T}_{[r]}^{\mathsf{H}}) \backslash \{\boldsymbol{0}\}$. Since $\boldsymbol{c}_r^{(1)} \boxtimes \boldsymbol{b}_r^{(1)} \not\perp \boldsymbol{x}_r^{(0)}$, it follows that $\boldsymbol{c}_r^{(1)} \boxtimes \boldsymbol{b}_r^{(1)} \notin \ker(\boldsymbol{T}_{[r]})$. The same reasoning can be applied to the next iterations, and the proof is complete. □

Lemma 5.4.2 *Let $\boldsymbol{a}_r^{(k)} = \boldsymbol{T}_{[r]}^{\mathsf{H}+}(\boldsymbol{c}_r^{(k)} \boxtimes \boldsymbol{b}_r^{(k)})$ be the minimal norm solution at k-th iteration in CAPD algorithm. Then for all $k \geq 0$, $\boldsymbol{a}_r^{(k)}$ is a linear combination of the columns of $\boldsymbol{A}$.*

Proof: It follows directly from the mode-1 unfolding $\boldsymbol{T} = \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^{\mathsf{T}}$ that the columns of $\boldsymbol{A}$ span $\mathrm{Ran}(\boldsymbol{T})$. Yet, $\boldsymbol{a}_r^{(k)} \in \mathrm{Ran}(\boldsymbol{T}_{[r]}^{\mathsf{H}+}) = \mathrm{Ran}(\boldsymbol{T}_{[r]}) \subseteq \mathrm{Ran}(\boldsymbol{T})$. □

Theorem 5.4.3 *Let $\boldsymbol{\mathcal{T}}$ be a tensor whose factor $\boldsymbol{A}$ is a column-wise orthogonal matrix. Assume that $\boldsymbol{\mathcal{T}}$ has an essentially unique decomposition, and $\boldsymbol{c}_r^{(0)} \boxtimes \boldsymbol{b}_r^{(0)} \notin \ker(\boldsymbol{T}_{[r]})$, $1 \leq r \leq R-1$. If $\boldsymbol{x}_r^{(k)}$, $k \geq 0$, converges to a point $\boldsymbol{x}_r^\star \in \mathcal{M}_r \cap \mathcal{N}$ in CAPD algorithm, then $\widehat{\boldsymbol{a}}_r \otimes \widehat{\boldsymbol{b}}_r \otimes \widehat{\boldsymbol{c}}_r$ is one of the rank-1 components of $\boldsymbol{\mathcal{T}}$.*

Proof: From 5.4.1 $\boldsymbol{x}_r^\star \neq \mathbf{0}$ because $\boldsymbol{c}_r^{(0)} \boxtimes \boldsymbol{b}_r^{(0)} \notin \ker(\boldsymbol{T}_{[r]})$, which implies that $\widehat{\boldsymbol{a}}_r \neq \mathbf{0}$, $\widehat{\boldsymbol{b}}_r \neq \mathbf{0}$, and $\widehat{\boldsymbol{c}}_r \neq \mathbf{0}$. From Lemma 5.4.2, $\widehat{\boldsymbol{a}}_r$ can be written as a linear combination of the columns of the factor $\boldsymbol{A}$. That is, $\widehat{\boldsymbol{a}}_r = \sum_{i=1}^{R} \alpha_i \boldsymbol{a}_i$. Since $\boldsymbol{x}_r^{(k)} \to \boldsymbol{x}_r^\star \in \mathcal{M}_r \cap \mathcal{N}$, it follows that $\widehat{\boldsymbol{a}}_r, \widehat{\boldsymbol{b}}_r$ and $\widehat{\boldsymbol{c}}_r$ satisfy equation (5.3), so that $\boldsymbol{x}_r^\star = \widehat{\boldsymbol{c}}_r^* \boxtimes \widehat{\boldsymbol{b}}_r^*$. Substituting $\widehat{\boldsymbol{a}}_r$ in (5.3)

$$\boldsymbol{T}^{\mathsf{H}} \widehat{\boldsymbol{a}}_r = \widehat{\boldsymbol{c}}_r^* \boxtimes \widehat{\boldsymbol{b}}_r^* \implies \sum_{i=1}^{R} \alpha_i \boldsymbol{T}^{\mathsf{H}} \boldsymbol{a}_i = \widehat{\boldsymbol{c}}_r^* \boxtimes \widehat{\boldsymbol{b}}_r^* \implies \widehat{\boldsymbol{c}}_r \boxtimes \widehat{\boldsymbol{b}}_r = \sum_{i=1}^{R} \alpha_i^* (\boldsymbol{c}_i \boxtimes \boldsymbol{b}_i).$$

Now, let $\alpha_j \neq 0$ for some $j \in \{1, 2, \ldots, R\}$, and write

$$\boldsymbol{c}_j \boxtimes \boldsymbol{b}_j = \frac{1}{\alpha_j^*} \left(\widehat{\boldsymbol{c}}_r \boxtimes \widehat{\boldsymbol{b}}_r - \sum_{\substack{i=1 \\ i \neq j}}^{R} \alpha_i^* (\boldsymbol{c}_i \boxtimes \boldsymbol{b}_i) \right).$$

Yet, a decomposition of $\boldsymbol{\mathcal{T}}$ along the mode-1 is given by

$$\boldsymbol{T} = \boldsymbol{a}_1 (\boldsymbol{c}_1 \boxtimes \boldsymbol{b}_1)^{\mathsf{T}} + \cdots + \boldsymbol{a}_R (\boldsymbol{c}_R \boxtimes \boldsymbol{b}_R)^{\mathsf{T}}.$$

Thus, plugging the expression of $\boldsymbol{c}_j \boxtimes \boldsymbol{b}_j$ into that of $\boldsymbol{T}$ and reorganizing the factors we obtain

$$\begin{aligned} \boldsymbol{T} = &\left(\boldsymbol{a}_1 - \frac{\alpha_1^*}{\alpha_j^*} \boldsymbol{a}_j \right) (\boldsymbol{c}_1 \boxtimes \boldsymbol{b}_1)^{\mathsf{T}} + \cdots + \left(\boldsymbol{a}_{j-1} - \frac{\alpha_{j-1}^*}{\alpha_j^*} \boldsymbol{a}_j \right) (\boldsymbol{c}_{j-1} \boxtimes \boldsymbol{b}_{j-1})^{\mathsf{T}} + \frac{1}{\alpha_j^*} \boldsymbol{a}_j (\widehat{\boldsymbol{c}}_r \boxtimes \widehat{\boldsymbol{b}}_r)^{\mathsf{T}} + \\ &+ \left(\boldsymbol{a}_{j+1} - \frac{\alpha_{j+1}^*}{\alpha_j^*} \boldsymbol{a}_j \right) (\boldsymbol{c}_{j+1} \boxtimes \boldsymbol{b}_{j+1})^{\mathsf{T}} + \cdots + \left(\boldsymbol{a}_R - \frac{\alpha_R^*}{\alpha_j^*} \boldsymbol{a}_j \right) (\boldsymbol{c}_R \boxtimes \boldsymbol{b}_R)^{\mathsf{T}}. \end{aligned}$$

It is important to mention that $\widehat{\boldsymbol{c}}_r \boxtimes \widehat{\boldsymbol{b}}_r$ is not in the column space of the vectors $\{\boldsymbol{c}_1 \boxtimes \boldsymbol{b}_1, \ldots, \boldsymbol{c}_{j-1} \boxtimes \boldsymbol{b}_{j-1}, \boldsymbol{c}_{j+1} \boxtimes \boldsymbol{b}_{j+1}, \ldots, \boldsymbol{c}_R \boxtimes \boldsymbol{b}_R\}$, otherwise $\operatorname{rank}(\boldsymbol{\mathcal{T}}) < R$.

Since the first factor matrix is column-wise orthogonal and the decomposition is essentially unique, the vectors

$$\{\boldsymbol{a}_1 - \frac{\alpha_1^*}{\alpha_j^*} \boldsymbol{a}_j, \ldots, \boldsymbol{a}_{j-1} - \frac{\alpha_{j-1}^*}{\alpha_j^*} \boldsymbol{a}_j, \frac{1}{\alpha_j^*} \boldsymbol{a}_j, \boldsymbol{a}_{j+1} - \frac{\alpha_{j+1}^*}{\alpha_j^*} \boldsymbol{a}_j, \ldots, \boldsymbol{a}_R - \frac{\alpha_R^*}{\alpha_j^*} \boldsymbol{a}_j\}$$

are orthogonal to one another. In particular, for $i \in \{1, 2, \ldots, R\} - \{j\}$

$$\left(\boldsymbol{a}_i - \frac{\alpha_i^*}{\alpha_j^*} \boldsymbol{a}_j \right)^{\mathsf{H}} \left(\frac{1}{\alpha_j^*} \boldsymbol{a}_j \right) = 0 \implies \alpha_i = 0,$$

which implies that $\widehat{\boldsymbol{a}}_r = \alpha_j \boldsymbol{a}_j$ and $\widehat{\boldsymbol{c}}_r \boxtimes \widehat{\boldsymbol{b}}_r = \alpha_j^* (\boldsymbol{c}_j \boxtimes \boldsymbol{b}_j)$. As $\widehat{\boldsymbol{c}}_r \boxtimes \widehat{\boldsymbol{b}}_r$ is a unit vector, it follows that $|\alpha_j| = 1 / \|\boldsymbol{c}_j \boxtimes \boldsymbol{b}_j\|$.

To eliminate scalar indeterminances, assume $\|\boldsymbol{c}_j \boxtimes \boldsymbol{b}_j\| = 1$, which leads to $|\alpha_j| = 1$. Thus,

$$\widehat{\boldsymbol{a}}_r \otimes \widehat{\boldsymbol{b}}_r \otimes \widehat{\boldsymbol{c}}_r = |\alpha_j| \boldsymbol{a}_j \otimes \boldsymbol{b}_j \otimes \boldsymbol{c}_j = \boldsymbol{a}_j \otimes \boldsymbol{b}_j \otimes \boldsymbol{c}_j,$$

and the proof is complete. □

Corollary 5.4.4 *$\boldsymbol{x}_r^\star \in \mathcal{M}_r \cap \mathcal{N} \implies \boldsymbol{x}_r^\star$ is a column of $\boldsymbol{C}^* \odot \boldsymbol{B}^*$.*

Proof: Up to scalar elementary indeterminances, it follows from Theorem 5.4.3 that $\boldsymbol{x}_r^\star = \widehat{\boldsymbol{c}}_r^* \boxtimes \widehat{\boldsymbol{b}}_r^* = \alpha_j(\boldsymbol{c}_j^* \boxtimes \boldsymbol{b}_j^*)$, for some $j \in \{1, 2, \ldots, R\}$. □

Theorem 5.4.3 ensures that one rank-1 component of tensor $\boldsymbol{\mathcal{T}}$ can be extracted if CAPD converges to a limit point $\boldsymbol{x}_r^\star \in \mathcal{M}_r \cap \mathcal{N}$. Actually, the convergence is not globally ensured since $\mathcal{N}$ is a non-convex manifold, which means that CAPD can get stuck if the starting point is badly chosen. Therefore, we focus our convergence study of CAPD method on starting points close to $\mathcal{M}_r \cap \mathcal{N}$, in which case we can draw some important results in the real field. Before, we introduce two basic definitions on manifolds.

Definition 1 *[Lewis 2008] In some space $\mathcal{E}$, let $\mathcal{A}$ and $\mathcal{B}$ be two C^k-manifolds around a point $\boldsymbol{x} \in \mathcal{A} \cap \mathcal{B}$. The manifolds $\mathcal{A}$ and $\mathcal{B}$ are transverse at $\boldsymbol{x}$ if*

$$T_{\mathcal{A}}(\boldsymbol{x}) + T_{\mathcal{B}}(\boldsymbol{x}) = \mathcal{E},$$

where $T_{\mathcal{A}}(\boldsymbol{x})$ and $T_{\mathcal{B}}(\boldsymbol{x})$ are the tangent spaces to $\mathcal{A}$ and $\mathcal{B}$ at $\boldsymbol{x}$, respectively.

Definition 2 *[Andersson 2013] In some space $\mathcal{E}$, let $\mathcal{A}$ and $\mathcal{B}$ be two C^k-manifolds around a point $\boldsymbol{x} \in \mathcal{A} \cap \mathcal{B}$. $\boldsymbol{x}$ is a non-tangential point if and only if*

$$T_{\mathcal{A}}(\boldsymbol{x}) \cap T_{\mathcal{B}}(\boldsymbol{x}) = T_{\mathcal{A}\cap\mathcal{B}}(\boldsymbol{x}),$$

where $T_{\mathcal{A}}(\boldsymbol{x})$, $T_{\mathcal{B}}(\boldsymbol{x})$ and $T_{\mathcal{A}\cap\mathcal{B}}(\boldsymbol{x})$ are the tangent spaces to $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{A} \cap \mathcal{B}$ at $\boldsymbol{x}$, respectively

According to [Lewis 2008], if two smooth manifolds are transverse, and the initialization is close enough to their intersection, then the alternating projection method converges linearly to a point at the intersection of the manifolds. This is stated in the following theorem

Theorem 5.4.5 *[Lewis 2008] In some space $\mathcal{E}$, let $\mathcal{A}$ and $\mathcal{B}$ be two transverse manifolds around a point $\bar{\boldsymbol{x}} \in \mathcal{A} \cap \mathcal{B}$. If the initial point $\boldsymbol{x}_0 \in \mathcal{E}$ is close to $\bar{\boldsymbol{x}}$, then the method of alternating projections*

$$\boldsymbol{x}^{(k+1)} = \pi_{\mathcal{A}}\pi_{\mathcal{B}}(\boldsymbol{x}^{(k)}),\ (k = 0, 1, 2, \ldots)$$

is well-defined, and the distance $d_{\mathcal{A}\cap\mathcal{B}}(\boldsymbol{x}_k)$ from the iterate $\boldsymbol{x}_k$ to the intersection $\mathcal{A} \cap \mathcal{B}$ decreases Q-linearly[3] to zero.

In our case, $\mathcal{M}_r$ are linear subspaces and $\mathcal{N}$ is locally smooth in the Euclidean space (see Example 2 in [Lewis 2008]). Thus, in particular conditions, the manifolds $\mathcal{M}_r$ and $\mathcal{N}$ are transverse for every $r \in \{1, 2, \ldots, R-1\}$. This is shown in the following.

Lemma 5.4.6 *Let $\mathcal{R} = \{\boldsymbol{x} \in \mathbb{R}^{JK} : \boldsymbol{x} = \boldsymbol{c} \boxtimes \boldsymbol{b}, \forall \boldsymbol{b} \in \mathbb{R}^J \setminus \{\boldsymbol{0}\}$ and $\forall \boldsymbol{c} \in \mathbb{R}^K \setminus \{\boldsymbol{0}\}\}$. Then $\forall \boldsymbol{x} \in \mathcal{R}$ it follows that $T_{\mathcal{R}}(\boldsymbol{x}) = T_{\mathcal{N}}(\boldsymbol{x}/\|\boldsymbol{x}\|)$.*

[3]The distance decreases Q-linearly to zero is $\lim_{k\to\infty} \|\boldsymbol{x}_k - \boldsymbol{x}^\star\|/\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^\star\| = 0$, for some $\boldsymbol{x}^\star \in \mathcal{A} \cap \mathcal{B}$.

Proof: Note that $\mathcal{R}$ can be viewed as the set of $J \times K$ rank-1 matrices. According to [Lewis 2008], the tangent space of $\mathcal{R}$ is given by

$$T_{\mathcal{R}}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{R}^{JK} : \boldsymbol{u}_i^{\mathsf{T}} \operatorname{Unvec}(\boldsymbol{y})\boldsymbol{v}_j = 0, \ \ \forall 1 < i \leq J, 1 < j \leq K\}$$

where $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_J\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_K\}$ are the sets of left and right singular vectors of the matrix $\operatorname{Unvec}(\boldsymbol{x}) = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathsf{T}}, \ \sigma_1 > 0$.

Define $\mathcal{U} = \{\boldsymbol{x} \in \mathbb{R}^{JK} : \|\boldsymbol{x}\| = 1\}$. Since the tangent space does not depend on the singular value σ_1, and $\mathcal{R} \cap \mathcal{U} = \mathcal{N}$, it follows that $T_{\mathcal{R}}(\boldsymbol{x}) = T_{\mathcal{R}}(\boldsymbol{x}/\|\boldsymbol{x}\|) = T_{\mathcal{N}}(\boldsymbol{x}/\|\boldsymbol{x}\|)$. □

Proposition 5.4.7 *Let $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$ be a rank-R tensor with factor $\boldsymbol{A}$ column-wise orthogonal. Assume neither $\boldsymbol{C}$ nor $\boldsymbol{B}$ have collinear columns. In CAPD algorithm, $\mathcal{M}_r$ and $\mathcal{N}$ are transverse manifolds at any point $\boldsymbol{x} \in \mathcal{M}_r \cap \mathcal{N}$ if $r \leq R - (J-1)(K-1)$, for $r \in \{1, 2, \ldots, R-1\}$.*

Proof: Let $\boldsymbol{x} \in \mathcal{M}_r \cap \mathcal{N}$, for some $r \in \{1, 2, \ldots, R-1\}$. According to Lemma 5.4.6, the tangent space of $\mathcal{N}$ at $\boldsymbol{x}$ is given by

$$T_{\mathcal{N}}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{R}^{JK} : \boldsymbol{u}_i^{\mathsf{T}} \operatorname{Unvec}(\boldsymbol{y})\boldsymbol{v}_j = 0, \ \ \forall 1 < i \leq J, 1 < j \leq K\},$$

for left and right singular vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_J\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_K\}$ of the matrix $\operatorname{Unvec}(\boldsymbol{x}) = \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathsf{T}}$ of dimension $J \times K$.

Let $\mathcal{K}(\boldsymbol{x}) = \{\operatorname{vec}(\boldsymbol{u}_1\boldsymbol{v}_1^{\mathsf{T}}), \operatorname{vec}(\boldsymbol{u}_1\boldsymbol{v}_2^{\mathsf{T}}), \ldots, \operatorname{vec}(\boldsymbol{u}_1\boldsymbol{v}_K^{\mathsf{T}}), \operatorname{vec}(\boldsymbol{u}_2\boldsymbol{v}_1^{\mathsf{T}}), \ldots, \operatorname{vec}(\boldsymbol{u}_J\boldsymbol{v}_1^{\mathsf{T}})\}$. Note that all vectors in $\mathcal{K}(\boldsymbol{x})$ are orthogonal to one another so that they span a $J + K - 1$ subspace in $\mathbb{R}^{JK}$.

Let also $\mathcal{D}(\boldsymbol{x}) = \{\boldsymbol{z} \in \mathbb{R}^{JK} : \boldsymbol{z} = \operatorname{vec}(\boldsymbol{u}_i\boldsymbol{v}_j^{\mathsf{T}}), \ \ \forall 1 < i \leq J, 1 < j \leq K\}$. Clearly, $\mathcal{D}(\boldsymbol{x}) \cap T_{\mathcal{N}}(\boldsymbol{x}) = \emptyset$. Actually, $\mathcal{D}(\boldsymbol{x})$ is a set of orthogonal vectors that is the complement of $T_{\mathcal{N}}(\boldsymbol{x})$. Indeed, $\mathcal{D}(\boldsymbol{x}) + \mathcal{K}(\boldsymbol{x}) = \mathbb{R}^{JK} \implies \mathcal{D}(\boldsymbol{x}) + T_{\mathcal{N}}(\boldsymbol{x}) = \mathbb{R}^{JK}$ because $\mathcal{K}(\boldsymbol{x}) \subseteq T_{\mathcal{N}}(\boldsymbol{x})$. Thus, we can conclude that $J + K - 1$ is the dimension of $T_{\mathcal{N}}(\boldsymbol{x}) = \operatorname{Span}\{\mathcal{K}(\boldsymbol{x})\}$.

Now, define $\mathcal{H}$ the set composed of the possible columns of $\boldsymbol{C} \odot \boldsymbol{B}$. Since neither $\boldsymbol{C}$ nor $\boldsymbol{B}$ have collinear columns,

$$\Big\{\mathcal{K}(\boldsymbol{x}) - \{\operatorname{vec}(\boldsymbol{u}_1\boldsymbol{v}_1^{\mathsf{T}})\}\Big\} \cap \mathcal{H} = \emptyset,$$

and thereby $\mathcal{K}(\boldsymbol{x}) - \{\operatorname{vec}(\boldsymbol{u}_1\boldsymbol{v}_1^{\mathsf{T}})\} \not\subseteq \mathcal{M}_r = T_{\mathcal{M}_r}(\boldsymbol{x})$.

Note that $\operatorname{vec}(\boldsymbol{u}_1\boldsymbol{v}_1^{\mathsf{T}})$ is the only vector of $\mathcal{K}(\boldsymbol{x})$ in $T_{\mathcal{M}_r}(\boldsymbol{x})$ so that any linear combination of at least two vectors in $\mathcal{K}(\boldsymbol{x})$ does not lie in $T_{\mathcal{M}_r}(\boldsymbol{x})$. In other words, $\operatorname{Span}\{\mathcal{K}(\boldsymbol{x})\} \cap T_{\mathcal{M}_r}(\boldsymbol{x}) = T_{\mathcal{N}}(\boldsymbol{x}) \cap T_{\mathcal{M}_r}(\boldsymbol{x}) = \{\beta \operatorname{vec}(\boldsymbol{u}_1\boldsymbol{v}_1^{\mathsf{T}})\}$ for $\beta \in \mathbb{R}$. Yet, the dimension of $T_{\mathcal{M}_r}(\boldsymbol{x})$ is equal to $R - r + 1$ since $\operatorname{rank}(\boldsymbol{T}_{[r]}^{\mathsf{T}}\boldsymbol{T}_{[r]}^{\mathsf{T}+}) = R - r + 1$. Hence,

$$\begin{aligned}\dim\{T_{\mathcal{N}}(\boldsymbol{x}) + T_{\mathcal{M}_r}(\boldsymbol{x})\} &= \dim\{T_{\mathcal{N}}(\boldsymbol{x})\} + \dim\{T_{\mathcal{M}_r}(\boldsymbol{x})\} - \dim\{T_{\mathcal{N}}(\boldsymbol{x}) \cap T_{\mathcal{M}_r}(\boldsymbol{x})\} \\ &= J + K + R - 1 - r.\end{aligned}$$

Thus, in order to ensure transversality between the manifolds, we should have $J + K + R - 1 - r \geq JK \implies r \leq R - (J-1)(K-1)$. □

Although Proposition 5.4.7 ensures the transversality between the manifolds $\mathcal{M}_r$ and $\mathcal{N}$ and, thus, the linear convergence of our algorithm, the condition $r \leq R - (J-1)(K-1)$ is very restrictive.

Indeed, to ensure the convergence of CAPD algorithm when $r = R - 1$, we must have $R - 1 \leq R - (J-1)(K-1) \implies (J-1)(K-1) \leq 1$, which is true only when $J \leq 2, K \leq 2$. In other cases, Proposition 5.4.7 can only ensure, at best, the convergence for a few values of r. For some scenarios, Table 5.1 shows how many rank-1 components can be ensured by CAPD using the transversality concept.

(J, K)	R	# of rank-1 components
$(2, 2)$	2	2
$(3, 3)$	2	0
$(3, 3)$	3	0
$(5, 5)$	3	0
$(5, 5)$	20	4
$(6, 8)$	20	0
$(6, 8)$	40	5
$(10, 15)$	100	0
$(20, 20)$	100	0

Table 5.1: Number of estimated components ensured by CAPD under the transversality concept.

Notice that both the dimension I (omitted in the table) and the rank R must be much larger than J and K in general, in order to estimate some rank-1 components with convergence guarantees. Even for rank-100 tensors with dimensions $I \times 20 \times 20$, with $I \geq 100$ for ensuring the column-wise orthogonality of factor $\boldsymbol{A}$, the convergence to any component cannot be ensured by our algorithm. Thus, the table evinces the limitation of Proposition 5.4.7.

Instead of considering transversality between manifolds, we can evaluate the convergence of CAPD under the non-tangential concept presented in Definition 2. According to [Andersson 2013], non-tangential points at the intersection of two manifolds is less restrictive than the transversality concept. Actually, the latter requires that the individual tangent spaces of the manifolds in a space must generate the whole space, which is not necessary for non-tangential points. The authors show that the convergence to a point at the intersection of two smooth manifolds can be ensured if the initialization of the alternating projection algorithm is close to a non-tangential point in the intersection. This result is shown in theorem below.

Theorem 5.4.8 *[Andersson 2013] Let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{A} \cap \mathcal{B}$ be C^2-manifolds, and let $\boldsymbol{x} \in \mathcal{A} \cap \mathcal{B}$ be a non tangential intersection point of $\mathcal{A}$ and $\mathcal{B}$. Given $\varepsilon > 0$ and $\cos(\alpha(\boldsymbol{x})) < c < 1$, where $\alpha(\boldsymbol{x})$ is the minimal angle between $\mathcal{A}$ and $\mathcal{B}$ at $\boldsymbol{x}$, there exists a $\rho > 0$ such that for any $\boldsymbol{x}^{(0)} \in \mathcal{B}_\rho(\boldsymbol{x})$, where $\mathcal{B}_\rho(\boldsymbol{x})$ is the open ball of radius ρ around $\boldsymbol{x}$, the sequence of alternating projections*

$$\boldsymbol{x}^{(k+1)} = \pi_{\mathcal{A}}\pi_{\mathcal{B}}(\boldsymbol{x}^{(k)}),\ (k = 0, 1, 2, \ldots)$$

(i) converges to a point $\boldsymbol{x}^\star \in \mathcal{A} \cap \mathcal{B}$,

(ii) $\|\boldsymbol{x}^\star - \pi_{\mathcal{A}\cap\mathcal{B}}(\boldsymbol{x}^{(0)})\| \leq \varepsilon\|\boldsymbol{x}^{(0)} - \pi_{\mathcal{A}\cap\mathcal{B}}(\boldsymbol{x}^{(0)})\|$,

(iii) $\|\boldsymbol{x}^\star - \boldsymbol{x}^{(k)}\| \leq \mathrm{cte} \cdot c^k\|\boldsymbol{x}^{(0)} - \pi_{\mathcal{A}\cap\mathcal{B}}(\boldsymbol{x}^{(0)})\|$.

We are mainly interested in the result *(iii)*. Indeed, $\boldsymbol{x}^{(k)} \to \boldsymbol{x}^\star$ as $k \to \infty$. In order to apply the result of [Andersson 2013] to our case, we must first prove under some conditions that there exists at least a non-tangential point $\boldsymbol{x}_r \in \mathcal{M}_r \cap \mathcal{N}$, for every $r \in \{1, 2, \ldots, R-1\}$. This is shown in the following proposition.

Proposition 5.4.9 *Let $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$ be a rank-R tensor with factor $\boldsymbol{A}$ column-wise orthogonal. If $\boldsymbol{\mathcal{T}}$ has an essentially unique decomposition then any point $\boldsymbol{x}_r \in \mathcal{M}_r \cap \mathcal{N}$, $r \in \{1, 2, \ldots, R-1\}$, is non-tangential.*

Proof: Since the decomposition is essentially unique, the columns of $\boldsymbol{B}$ and $\boldsymbol{C}$ are unique up to scalar factors and not collinear, otherwise the Kruskal's rank of some factor matrix would equal to 1, implying non-uniqueness. Moreover, from Collorary 5.4.4, all points in $\mathcal{M}_r \cap \mathcal{N}$ are columns of $\boldsymbol{C} \odot \boldsymbol{B}$. Thus, it turns out that $\exists \rho > 0 : \mathcal{B}_\rho(\boldsymbol{x}_r) \cap \mathcal{M}_r \cap \mathcal{N} = \{\beta \boldsymbol{x}_r\},\ \forall \beta \in \mathbb{R}$ such that $|\beta| < \rho$. Since the straight along $\boldsymbol{x}_r$ is a linear space, we can conclude that $T_{\mathcal{M}_r \cap \mathcal{N}}(\boldsymbol{x}_r) = \{\gamma \boldsymbol{x}_r\},\ \forall \gamma \in \mathbb{R}$.

On the other hand, the uniqueness of $\boldsymbol{x}_r \in \mathcal{M}_r \cap \mathcal{N}$ within $\mathcal{B}_\rho(\boldsymbol{x}_r)$ ensures that $T_{\mathcal{N}}(\boldsymbol{x}_r) = \{\gamma \boldsymbol{x}_r\},\ \forall \gamma \in \mathbb{R}$, which leads to $T_{\mathcal{N}}(\boldsymbol{x}_r) \cap T_{\mathcal{M}_r}(\boldsymbol{x}_r) = \{\gamma \boldsymbol{x}_r\},\ \forall \gamma \in \mathbb{R}$, since $T_{\mathcal{M}_r}(\boldsymbol{x}_r) = \mathcal{M}_r$. Hence, $T_{\mathcal{M}_r}(\boldsymbol{x}_r) \cap T_{\mathcal{N}}(\boldsymbol{x}_r) = T_{\mathcal{M}_r \cap \mathcal{N}}(\boldsymbol{x}_r)$, at any point $\boldsymbol{x}_r \in \mathcal{M}_r \cap \mathcal{N}$, for every $r \in \{1, 2, \ldots, R-1\}$. □

Proposition 5.4.9 brings up a strong result that ensures the convergence of CAPD algorithm. The only imposed constraint is the uniqueness of the decomposition, which is desired in practical applications.

5.5 Results on the orthogonal CP decomposition

In this section, we evaluate the performance of CAPD algorithm by numerical experiments. First, we show how the distance between iterates of the alternating projection part of our algorithm decreases as the number of iterations increases. Second, we will see that the convergence of a random tensor with a column-wise orthogonal factor matrix is log-linear. Finally, we present the advantage of CAPD with respect to the alternating projection method that computes the factor matrices all-at-once.

5.5.1 Iterates of CAPD algorithm for real tensors

We consider a sample of 1000 real tensors with the factor matrices generated as follows

- The entries of $\boldsymbol{B}$ and $\boldsymbol{C}$ are distributed according to a uniform measure in $[-1, 1]$;
- For $\boldsymbol{A}$, we first generate a random matrix with entries obeying a uniform distributed in $[-1, 1]$, and then set $\boldsymbol{A}$ as the left singular matrix of that random matrix, ensuring the orthogonality of the factor.

Let $\Delta_k = \|\boldsymbol{x}_r^{(k+1)} - \boldsymbol{x}_r^{(k)}\|$ be the norm of the difference between two successive iterates in CAPD algorithm for some $1 \leq r \leq R-1$ at iteration $k \geq 0$. Since we have 1000 tensors, we replace $\mathbb{E}\{\Delta_k\}$ by the average value at iteration k. We assume the tensors have rank 4, so that we evaluate the

algorithm by estimating 3 rank-1 components (the last component is obtained directly by performing the deflation, as shown in Algorithm 7) for four scenarios.

Figure 5.1 shows for the scenarios where $K = 3$ that the average distance between iterates decreases as the rank of the tensor is reduced by deflation. For the scenario $20 \times 20 \times 20$, the same phenomenon is noted for $10^{-2} < \mathbb{E}\{\Delta_k\} < 10^{-6}$, which covers most of the range of the other scenarios. We also note from the last scenario that for tensors with all dimensions much larger than the rank, the CAPD algorithm converges in a few iterations. Indeed, the iterates converge to zero[4] in 60, 55, and 40 iterations approximately,for r equal to $1, 2$ and 3, respectively.

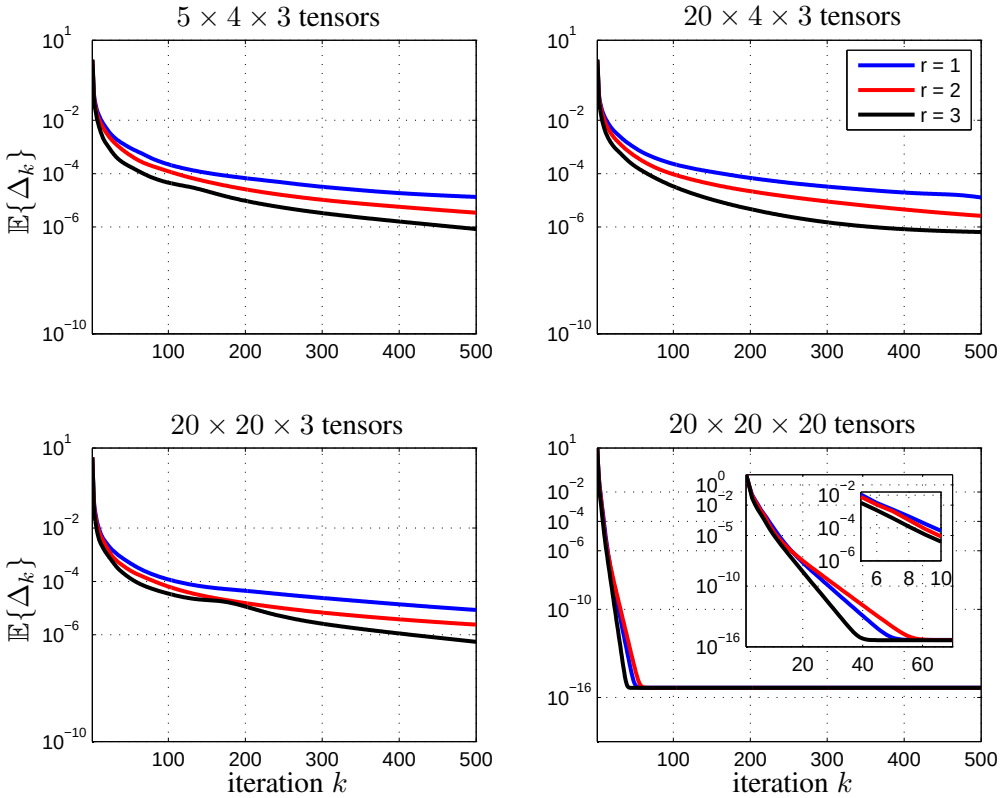


Figure 5.1: Convergence of the iterates of CAPD algorithm for rank-4 tensors.

Now, we pick randomly one tensor of the sample to show how the iterates converge as k increases. We choose the scenario $5 \times 4 \times 3$. In Figure 5.2, we see that the distance of iterates of CPAD algorithm converges log-linearly to zero. Although we do not show other examples here, this behavior can be noted for most of the tensors of our sample. Moreover, the log-linear convergence also depends on the starting point $\boldsymbol{x}_r^{(0)}$.

Contrary to the averaged result, we note that $\|\boldsymbol{x}_r^{(k+1)} - \boldsymbol{x}_r^{(k)}\|$, for some k, does not decrease when the rank decreases. Indeed, the convergence is slower when the rank of the tensor is only 2 ($r = 3$).

[4]It is the virtual zero or the floating-point relative accuracy of Matlab, whose value is 2^{-52}.

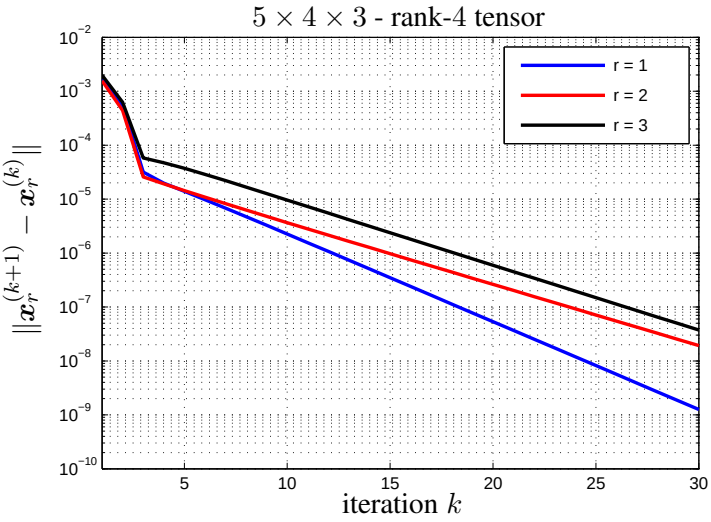


Figure 5.2: Convergence of the iterates of CAPD algorithm for a single rank-4 tensor.

5.5.2 Performance for complex tensors

Herein, we evaluate the performance of CAPD in terms of computational time and the percentage of successful decompositions denoted by $\%p$. We assume that CAPD delivers a correct decomposition for a tensor $\boldsymbol{\mathcal{T}}$ if $\|\boldsymbol{\mathcal{T}} - \widehat{\boldsymbol{\mathcal{T}}}\| \leq 10^{-6}$, where $\widehat{\boldsymbol{\mathcal{T}}}$ is the estimated tensor. We generate a sample of 500 complex tensors. The real and imaginary parts are uniformly distributed in $[-1, 1]$, and the column-wise factor $\boldsymbol{A}$ is the left singular matrix of a random complex matrix with uniform entries generated as before. We simulate the same four scenarios as in the previous subsection. Table below summarizes the results.

Scenario	Average time (seconds)	$\%p$
$5 \times 4 \times 3$	0.7231	99.6
$20 \times 4 \times 3$	0.7131	99
$20 \times 20 \times 3$	0.7942	100
$20 \times 20 \times 20$	0.2723	100

Table 5.2: Time and percentage of correct decompositions of CAPD for rank-4 tensors.

Notice that the average time of simulations for $20 \times 20 \times 20$ tensors corroborates with the convergence performance of iterates depicted in Figure 5.1, where we have shown that CAPD algorithm converges very fast when the dimensions are much larger than the rank. Here, it results in a small computational time compared to the other scenarios.

5.5.3 Simultaneous estimation of factors

As mentioned in Remark 5, the estimation of the factor matrices all-at-once might be performed by using an alternating projection algorithm based on equation (5.2). Instead of initializing the vector

$\boldsymbol{c}_r^{(0)} \boxtimes \boldsymbol{b}_r^{(0)}$, we could initialize directly a Khatri-Rao matrix $\boldsymbol{C}^{(0)} \odot \boldsymbol{B}^{(0)}$, and thus we would have the following alternating projection problem

$$\boldsymbol{X}^{(k+1)} = \pi_{\mathcal{M}} \pi_{\mathcal{N}_R}(\boldsymbol{X}^{(k)}),\ (k = 0, 1, 2, \ldots),$$

where

$$\mathcal{M} = \left\{ \boldsymbol{X} \in \mathbb{K}^{JK \times R} : \left(\boldsymbol{T}^{\mathsf{H}} \boldsymbol{T}^{\mathsf{H}+} - \boldsymbol{I} \right) \boldsymbol{X} = \boldsymbol{0} \right\},$$

and

$$\mathcal{N}_R = \{ \boldsymbol{X} \in \mathbb{K}^{JK \times R} : \boldsymbol{X} = \boldsymbol{C} \odot \boldsymbol{B}, \|\boldsymbol{X}\| = 1, \forall \boldsymbol{B} \in \mathbb{K}^{J \times R} \setminus \{\boldsymbol{0}\} \text{ and } \forall \boldsymbol{C} \in \mathbb{K}^{K \times R} \setminus \{\boldsymbol{0}\} \}.$$

If the algorithm converged to a point in $\mathcal{M} \cap \mathcal{N}_R$ so that the right factors $\boldsymbol{B}$ and $\boldsymbol{C}$ could be properly estimated, then the factor $\boldsymbol{A}$ would be estimated by solving the matrix linear equation (5.2), whose the least square solution is $\boldsymbol{A} = \boldsymbol{T}^{\mathsf{H}+}(\boldsymbol{C} \odot \boldsymbol{B})$.

This approach presents two crucial drawbacks:

- The complexity of this *all-at-once* alternating projection algorithm is larger than that of the CAPD algorithm. Indeed, the employment of an alternating projection algorithm to solve equation (5.2) is equivalent to employ R parallel alternating projection algorithms to solve R equations (5.3). Notice, however, that CAPD solves only $R - 1$ times the equation (5.3) .

- There is no guarantee that the estimated columns of the factors $\widehat{\boldsymbol{A}}$, $\widehat{\boldsymbol{B}}$, and $\widehat{\boldsymbol{C}}$ be the same of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$. As the initialization $\boldsymbol{C}^{(0)} \odot \boldsymbol{B}^{(0)}$ is actually R rank-1 initializations $\boldsymbol{c}_r^{(0)} \boxtimes \boldsymbol{b}_r^{(0)}$ (of R parallel problems), some of those initializations can converge to the same point, which means that the estimated matrices might have collinear columns. Indeed, this happens more often that one can imagine.

In order to show the limitation of computing the factors all-at-once using the alternating projection method, we count how many column factors are properly estimated for the matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$. We evaluate the following real and complex scenarios for a sample of 100 tensors: $15 \times 10 \times 5$, $20 \times 10 \times 5$, $20 \times 20 \times 5$ and $20 \times 20 \times 20$ tensors. For each tensor, we perform a random initialization.

Figure 5.3 shows that the estimation of all factors all-at-once using the alternating projection method happens in a very few cases. In fact, the best scenario is the one with $20 \times 20 \times 5$ real tensors, in which only for 7 tensors it was possible to extract completely the factor matrices. In all scenarios, the algorithm recovers 3 or 4 column factors of $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ in general. To increase the chances of estimating the right columns factors, we could test more initializations. However we would run again the algorithm, increasing though the number of operations to be computed. Therefore, CAPD is more efficient than the all-at-once alternating projection computation.

5.6 Chapter Summary and Directions

This chapter finishes with the following conclusions and further directions:

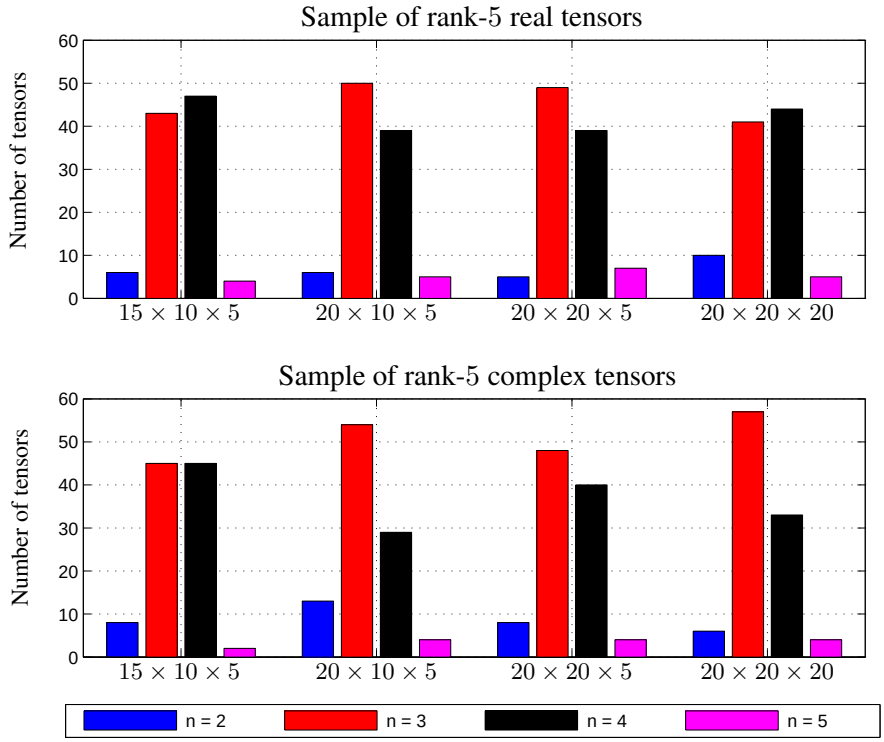


Figure 5.3: Tensors in which n column factors are correctly extracted with the *all-at-once* algorithm.

- *CAPD algorithm.* The alternating projection method combined with the deflation of rank-1 decomposable tensors arose as an alternative method to compute the exact CP decomposition of an orthogonal tensor with a semi-unitary factor matrix.

- *Extraction of factors with CAPD algorithm.* We proved that CAPD algorithm extracts one column of each factor matrix when the iterates of our proposed algorithm converges to a point in the intersection of two manifolds.

- *Transversality of points in CAPD algorithm.* Under some conditions on the factor matrices $\boldsymbol{B}$ and $\boldsymbol{C}$, we showed that the convergence of the alternating projection part of the CAPD algorithm to a point in $\mathcal{M}_r \cap \mathcal{N}$ is linear when the initialization of the algorithm is close to $\mathcal{M}_r \cap \mathcal{N}$ and the transversality at points in the intersection of the manifolds $\mathcal{M}_r$ and $\mathcal{N}$ is satisfied. In that case, we could evaluate how many columns factors can be extracted for different sizes and rank of tensors.

- *Non-tangential concept in CAPD algorithm.* In order to drop the restriction on the dimensions of tangent spaces of the manifolds under the transversality point of view, we used the concept of non-tangential points. In this case, we proved that CAPD algorithm converges to the exact decomposition when the initialization is close enough to $\mathcal{M}_r \cap \mathcal{N}$, for every $1 \leq r \leq R-1$.

- *Performance of CAPD algorithm.* From numerical simulations, we saw that the distance between iterates in the alternating projection part of CAPD decreases log-linearly to zero. We also showed that the computation of factors using CAPD performs better than the computation of factors all-at-once using an other alternating projection method applied directly to the

whole factor matrices. Indeed, it was shown that the complexity of CAPD is smaller and the *all-at-once* method does not ensure the exact estimation of the factor matrices.

- *Low approximation with CAPD*. For future work, we intend to evaluate CAPD algorithm under noisy scenarios. We already note that the deflation procedure compromises the estimation of the factor matrices since one of the estimated rank-1 factors is not an exact component of the noiseless tensor. We do not know exactly in which scenarios we can take advantage of CPDA when compared with other standard algorithms, such as ALS and those described in [Sørensen 2012].

Conclusions

Main conclusions

We summarize the main conclusions of this thesis.

Chapter 2

- The computation of the best rank-1 approximation using the moment approach due to Lasserre was compromised by the exponential growth of variables due to the problem relaxations, in spite of the method has provided a certificate of global optimality. This variable drawback limited simulations for real tensors of order 3 and dimension $2 \times 2 \times 2$. A test with tensors with dimension $3 \times 3 \times 3$ showed that the computational time to compute a best rank-1 approximation is approximately 30 minutes, which show the inefficency of the method.

- The proposed SeROAP algorithm is always at least as good as the ST-HOSVD algorithm (and consequently as THOSVD) for three-way tensors. The mathematical proof was confirmed by numerical experiments. Moreover, the computational complexity of SeROAP is smaller than that of THOSVD and it is of the same order of the complexity of the ST-HOSVD algorithm.

- The proposed CE is an iterative algorithm to compute rank-1 three-way tensor approximations. It presents small computational complexity than ALS algorithm when one dimension is sufficiently larger than the other two dimensions. We saw that the approximate error and the number of iterations needed to converge is smaller for CE in this scenario. We also presented some theoretical results ensuring that the objective function of the rank-1 approximation problem converges to a stationary value when CE is employed.

Chapter 3

- We proposed an iterative deflation algorithm to compute low rank tensor approximations whose mainly idea is to update rank-1 components using successive rank-1 approximations. This algorithm, called DCPD, is different to the method propose in [Cichocki 2009c] that updating the factors of the tensors using ALS. With our approach, we saw by simulations that residuals are quickly reduced, so that DCPD converges in a few iterations.

- We performed a study on the convergence of DCPD considering the best rank-1 approximation of the components within. Three main results were presented: (i) the monotonic decrease of a sequence of residuals; (ii) The recovering of an exact decomposition for DCPD when the residuals obeys some geometric conditions; (iii) a conjecture about the existence of continuous probability measures ensuring the convergence of DCPD with high probability.

Chapter 4

- In our thesis, we dedicated a chapter for tackling multivariate quadratic systems. We showed that this problem can be reduced to the best rank-1 approximation problem. Thus, some tensor tools, such as iterative algorithms, can be employed to extract solutions of a particular system. The reduction was proved for the real field. We also gave a list of advantages and drawbacks of our tensor approach.
- We generalized the tensor approach of the real case for a specific but more general complex system. We showed that our complex system can be reduced to a conjugated partially symmetric rank-1 approximation problem.
- To compute a conjugated partially symmetric rank-1 approximation of a tensor, we proposed an algorithm, called CPS, based on the stationary equations of (P9). We saw by numerical experiments, that this algorithm can extract one complex solution of the system.

Chapter 5

- we proposed an algorithm, called CAPD, combining the alternating projection and deflation methods to compute the exact decomposition of tensors with a column-wise orthogonal matrix.
- We showed that CAPD extracts all rank-1 components of the tensor if the iterates of the algorithm converge to points at the intersection of two known manifolds.
- If some transversality conditions are satisfied, we showed that the iterates of CAPD converge to a point at the intersection of manifolds. However, we saw that the transversality concept applied to our problem does not ensure the extraction of all rank-1 components of a tensor.
- To overcome the restrictive condition imposed by transversality, we used the non-tangential concept to show that the convergence of the CAPD algorithm can be ensured when starting points (initialization of the algorithm) are close to the intersection of predefined manifolds. Thus, the extraction of all rank-1 components is ensured.
- We also showed that the extraction of rank-1 components one-by-one is less computationally complex than the estimation of the factor matrices all-at-once.

Perspectives

In the following, we draw the main directions for future works:

Chapter 2

- The employment of other algebraic geometry tools such as [Bucero 2014] to compute best rank-1 approximations can outperform the Lasserre's relaxation method, but it still concerns tensors with small dimensions. The monomials of the objective of Problem 2.2 are sparse, so that we

expect to improve the computational time to compute the best rank-1 approximation taking into account their sparsity.

- SeROAP arose as a simple and efficient non-iterative algorithm to compute rank-1 approximations. In a future work, we intend to develop new strategies to tackle higher order tensors outperforming SeROAP. We also have some ideas to improve SeROAP, such as finding better approximations by looking for other rank-1 tensors from that one delivered by SeROAP.

- For the CE algorithm, we do not prove that it converges to a stationary point, but only that the objective of the best rank-1 approximation problem converges to some value. The proof is still a challenge.

Chapter 3

- The proof of Conjecture 3.2.8 is an open problem, even for some constrained cases (e.g. orthogonal tensors). We think that some tools in theory measure and algebraic geometry can gives us some directions to overcome this problem.

Chapter 4

- There are some unanswered questions about the reduction of polynomial quadratic systems into rank-1 approximation problems: (i) What are the conditions ensuring the equivalence between problems (P1) and (P2)? (ii) What is the set of tensors at which the equality $\mathcal{S}_{P4} = \mathcal{S}_{P5}$ holds? We intend to tackle these two question soon.

Chapter 5

- We do not evaluate the CAPD algorithm under noisy scenarios. Actually, the theoretical results only work for the exact CP decomposition. However, we intend to evaluate the impact of noise on the estimation of factor matrices. An idea for mitigating the errors due to noise is to approximate the unfolding matrix $\boldsymbol{\mathcal{T}}_{[r]}$ to a rank $R - r + 1$ matrix by computing the SVD. We do not know yet whether this will be able to improve the estimations.

A

Appendices

A.1 Iterative CP decomposition algorithms

This appendix presents the description of two iterative tensor decomposition algorithms used in our simulations: ALS [Smilde 2005, Comon 2009b] and HALS [Cichocki 2009c]. These algorithms are used to compute low rank CP tensor approximations.

In the following, the input parameter R denotes the rank of the output tensor. Assuming R_0 is the rank of the input tensor $\boldsymbol{\mathcal{T}}$, if $R_0 \leq R$, then the algorithms perform an exact decomposition. On the other hand, if $R_0 > R$, a lower rank-R approximation is computed.

Alternating least squares

The most commonly used algorithm for solving the CP decomposition is ALS [Smilde 2005]. The goal is to update alternately each factor matrix in each iteration by solving a least squares problem conditioned on previous updates of the other factor matrices. The implementation is quite simple and it is detailed in Algorithm 8.

input : $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: input data,
R: rank parameter.
output: $\boldsymbol{A}^{(n)} \in \mathbb{K}^{I_n \times R}$, for $n = 1, \ldots, N$: factor matrices
Initialize $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}$
repeat
  for $n = 1$ *to* N **do**
    $\boldsymbol{V} \leftarrow (\boldsymbol{A}^{(1)})^{\mathsf{T}}\boldsymbol{A}^{(1)} \boxdot \cdots \boxdot (\boldsymbol{A}^{(n-1)})^{\mathsf{T}}\boldsymbol{A}^{(n-1)} \boxdot (\boldsymbol{A}^{(n+1)})^{\mathsf{T}}\boldsymbol{A}^{(n+1)} \boxdot \cdots \boxdot (\boldsymbol{A}^{(N)})^{\mathsf{T}}\boldsymbol{A}^{(N)}$
    $\boldsymbol{A}^{(n)} \leftarrow \boldsymbol{T}^{(n)}(\boldsymbol{A}^{(N)} \odot \cdots \odot \boldsymbol{A}^{(n+1)} \odot \boldsymbol{A}^{(n-1)} \odot \cdots \odot \boldsymbol{A}^{(1)})\boldsymbol{V}^{+}$
  end
until *some stopping criterion is satisfied*;

Algorithm 8: ALS algorithm

Hierarchical ALS

The Hierarchical ALS algorithm was originally proposed in [Cichocki 2009c] in the context of non-negative matrix factorizations. Herein, we present its higher order version for complex tensors.

input : $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$: input data,
R: rank parameter.
output: $\boldsymbol{A}^{(n)} \in \mathbb{K}^{I_n \times R}$, for $n = 1, \ldots, N$: factor matrices
Initialize $\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}$;
$\boldsymbol{a}_r^{(n)} \leftarrow \boldsymbol{a}_r^{(n)} / \|\boldsymbol{a}_r^{(n)}\|$, $\forall r, n = 1, 2, \ldots, N-1$;
$\boldsymbol{\mathcal{E}} = \boldsymbol{\mathcal{T}} - \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \otimes \boldsymbol{a}_r^{(2)} \otimes \cdots \otimes \boldsymbol{a}_r^{(N)}$;
repeat
  for $r = 1$ *to* R **do**
    $\boldsymbol{\mathcal{T}}_{[r]} \leftarrow \boldsymbol{\mathcal{E}} + \boldsymbol{a}_r^{(1)} \otimes \boldsymbol{a}_r^{(2)} \otimes \cdots \otimes \boldsymbol{a}_r^{(N)}$;
    for $n = 1$ *to* N **do**
      $\boldsymbol{a}_r^{(n)} \leftarrow$
      $\boldsymbol{T}_{[r]}^{(n)} \left(\boldsymbol{a}_r^{(N)*} \otimes \boldsymbol{a}_r^{(N-1)*} \otimes \boldsymbol{a}_r^{(n+1)*} \otimes \cdots \otimes \boldsymbol{a}_r^{(n-1)*} \otimes \cdots \otimes \boldsymbol{a}_r^{(2)*} \otimes \boldsymbol{a}_r^{(1)*} \right)$;
      if $n \neq N$ **then**
        $\boldsymbol{a}_r^{(n)} \leftarrow \boldsymbol{a}_r^{(n)} / \|\boldsymbol{a}_r^{(n)}\|$;
      end
      $\boldsymbol{\mathcal{E}} \leftarrow \boldsymbol{\mathcal{T}}_{[r]} - \boldsymbol{a}_r^{(1)} \otimes \boldsymbol{a}_r^{(2)} \otimes \cdots \otimes \boldsymbol{a}_r^{(N)}$;
    end
  end
until *some stopping criterion is satisfied*;

Algorithm 9: HALS algorithm

A.2 Moment relaxation: a brief overview

Let the following optimization problem

$$p^{\min} = \inf_{\boldsymbol{x} \in \mathcal{K}} p(\boldsymbol{x}), \tag{A.1}$$

where $\mathcal{K} = \{\boldsymbol{x} \in \mathbb{R}^n : g_1(\boldsymbol{x}) \geq 0, g_2(\boldsymbol{x}) \geq 0, \ldots, g_m(\boldsymbol{x}) \geq 0\}$, with $p(\boldsymbol{x}), g_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$, for $i \in \{1, 2, \ldots, m\}$.

According to Lasserre's approach [Lasserre 2001], Problem (A.1) is equivalent to

$$p^{\min} = \inf_{\mu} \int_{\mathcal{K}} p(\boldsymbol{x}) \mu(dx), \tag{A.2}$$

where μ is a probability measure on $\mathbb{R}^n$ supported by the set $\mathcal{K}$. Note that $p(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}} p_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}}$, $\alpha \in \mathbb{N}^n$, so that it turns out that $\int p(\boldsymbol{x}) \mu(dx) = \boldsymbol{p}^T \boldsymbol{y}$, where $\boldsymbol{p}$ is the vector of coefficients $p_{\boldsymbol{\alpha}}$ abiding by some monomial order indexed by $\boldsymbol{\alpha}$, and $\boldsymbol{y}$ the sequence of moments of μ such that $y_{\boldsymbol{\alpha}} = \int \boldsymbol{x}^{\boldsymbol{\alpha}} \mu(dx)$. For more details about monomial orders refer to [Cox 1992], and on theory of moments see [Berg 1987, Curto 1991, Putinar 1999].

Problem (A.2) can be formulated as

$$\begin{aligned} p^{\min} = \inf_{\boldsymbol{y}} \boldsymbol{p}^T \boldsymbol{y} \\ s.t.\ y_0 = 1, \\ \boldsymbol{y} \text{ is a sequence of moments with a representing measure on } \mathcal{K}. \end{aligned} \tag{A.3}$$

For a sequence of moments, an infinite-dimensional moment matrix can be set up with entries indexed by $\mathbb{N}^n$, videlicet each element $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of the matrix is given by $y_{\boldsymbol{\alpha}+\boldsymbol{\beta}}$, for $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}^n$ [Lasserre 2001].

Let $\boldsymbol{M}(\boldsymbol{y})$ be the moment matrix for $\boldsymbol{y}$ and define $\boldsymbol{g}^{(\boldsymbol{y})} := \boldsymbol{M}(\boldsymbol{y})\boldsymbol{g}$, with $\boldsymbol{g}$ the sequence of coefficients of a polynomial $g(\boldsymbol{x})$ whose entries are sorted according to the same monomial order of $\boldsymbol{y}$. The $\boldsymbol{\alpha}$-entry of $\boldsymbol{g}^{(\boldsymbol{y})}$ is given by $(\boldsymbol{g}^{(\boldsymbol{y})})_{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\beta}} g_{\boldsymbol{\beta}} y_{\boldsymbol{\alpha}+\boldsymbol{\beta}}$. From Lemma 4.2 within [Laurent 2009], the last constraint in Problem (A.3) can be relaxed by the weak conditions $\boldsymbol{M}(\boldsymbol{y}) \succeq \boldsymbol{0}$ and $\boldsymbol{M}(\boldsymbol{g}_i^{(\boldsymbol{y})}) \succeq \boldsymbol{0}, \forall i \in \{1, 2, \ldots, m\}$, which leads to the following relaxed optimization problem whose global solution is a lower bound for $p^{\min}$.

$$\begin{aligned} p^{\text{mom}} = \inf_{\boldsymbol{y}} \boldsymbol{p}^T \boldsymbol{y} \\ s.t.\ y_0 = 1, \\ \boldsymbol{M}(\boldsymbol{y}) \succeq \boldsymbol{0}, \\ \boldsymbol{M}(\boldsymbol{g}_i^{(\boldsymbol{y})}) \succeq \boldsymbol{0}, \forall i \in \{1, 2, \ldots, m\}. \end{aligned} \tag{A.4}$$

Since it is not clear how to proceed with infinity matrices, the previous problem must be relaxed by truncating the involved moment matrices. For this, define $\boldsymbol{M}_t(\boldsymbol{y})$ the moment matrix for $\boldsymbol{y} \in \mathbb{N}^n_{2t}$ (the sequence of moments up to degree $2t$). Let $d_p = \lceil \deg(p(\boldsymbol{x}))/2 \rceil$, $d_{g_i} = \lceil \deg(g_i(\boldsymbol{x}))/2 \rceil$ and $d_{\mathcal{K}} = \max(d_{g_1}, d_{g_2}, \ldots, d_{g_m})$ (set $d_{\mathcal{K}} = 1$ if $m = 0$). For any integer $t \geq \max(d_p, d_{\mathcal{K}})$, it follows the sequence of SDP problems

$$\begin{aligned} p_t^{\text{mom}} = \inf_{\boldsymbol{y}} \boldsymbol{p}^T \boldsymbol{y} \\ s.t.\ y_0 = 1, \\ \boldsymbol{M}_t(\boldsymbol{y}) \succeq \boldsymbol{0}, \\ \boldsymbol{M}_{t-d_{g_i}}(\boldsymbol{g}_i^{(\boldsymbol{y})}) \succeq \boldsymbol{0}, \forall i \in \{1, 2, \ldots, m\}. \end{aligned} \tag{A.5}$$

One clearly notes that $p_t^{\text{mom}} \leq p^{\text{mom}} \leq p^{\min}$. Problems (A.5) can be viewed as moment hierarchical problems: as t gets larger, p_t^{mom} is closer to p^{mom}. The number of variables increases to $\binom{n+2t}{2t}$, as long as the lexicographic monomial order is considered.

Next, we present two known theoretical results on moment relaxations whose main ideas boil down to ensure an attainable infimum of Problems (A.5), and a certificate of optimality allowing to extract global minimizers of Problem (A.1).

Proposition A.2.1 *[Laurent 2009] If the ball constraint $R^2 - \sum_{i=1}^n x_i^2 \geq 0$ is present in the description of the semialgebraic set $\mathcal{K}$, then the feasible region of the relaxed problems (A.5) is bounded and its infimum is attained.*

Theorem A.2.2 *[Laurent 2009] Let* $t \geq \max(d_p, d_{\mathcal{K}})$ *and* $\mathcal{K}_p = \{\boldsymbol{x} \in \mathcal{K} | p(\boldsymbol{x}) = p^{min}\}$. *Let also* $\boldsymbol{y} \in \mathbb{N}^n_{2t}$ *be a global minimum of the problem (A.5). Assume* $\mathrm{card}(\mathcal{K}_p) < \infty$ *and* $\exists s$ *s.t.* $\max(d_p, d_{\mathcal{K}}) \leq s \leq t$ *and* $\mathrm{rank}(\boldsymbol{M}_s(\boldsymbol{y})) = \mathrm{rank}(\boldsymbol{M}_{s-d_K}(\boldsymbol{y}))$. *Then,* $p_t^{mom} = p^{min}$ *and* $V_{\mathbb{C}}(\ker(\boldsymbol{M}_s(\boldsymbol{y}))) \subseteq \mathcal{K}_p$.

To compute $V_{\mathbb{C}}(\ker(\boldsymbol{M}_s(\boldsymbol{y})))$, we can use the eigenvalue method in the multivariate case applied to the ideal constructed from a flat extension of $\boldsymbol{M}_s(\boldsymbol{y})$. More details is presented in [Henrion 2005].

The software GloptiPoly© [Henrion 2003, Henrion 2009] implements the moment problems (A.5) with the lexicographic basis, tests optimality and extracts global optimizers.

B

Résumé détaillé en français (extended abstract in French)

Ceci est un résumé détaillé en français des travaux réalisés durant ma thèse. D'abord nous présentons une courte introduction sur la décomposition polyadique canonique. Pour les autres chapitres, nous détaillerons les principales contributions en forme synthétique. À la fin, les perspectives de ce travail de thèse seront décrites.

Introduction

L'approximation tensorielle de rang faible joue ces dernières années un rôle important dans plusieurs applications, telles que la séparation aveugle de source, les télécommunications, le traitement d'antennes, les neurosciences, la chimiométrie, et la fouille de données. La décomposition tensorielle Polyadique Canonique (CP) est très attractive comparativement à des outils matriciels classiques, notamment pour l'identification de systèmes. En fait, l'approche tensorielle présente des propriétés intéressantes d'unicité sous des faibles contraintes, tandis qu'avec la SVD (un exemple d'outil matriciel largement utilisé) la contrainte d'orthogonalité doit être imposée pour assurer l'unicité de la décomposition.

Soit $\boldsymbol{\mathcal{T}} \in \mathbb{K}^{I_1 \times I_2 \times \cdots \times I_N}$ un tenseur d'ordre N avec ses composantes définies dans un corps $\mathbb{K}$. La décomposition CP est représentée par

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \lambda_r (\boldsymbol{a}_1^{(r)} \otimes \cdots \otimes \boldsymbol{a}_N^{(r)}), \tag{B.1}$$

où R est le nombre minimal de composantes de rang 1, aussi connue comme le rang du tenseur [Hitchcock 1927]. La Figure B.1 décrit la décomposition CP pour des tenseurs d'ordre 3.

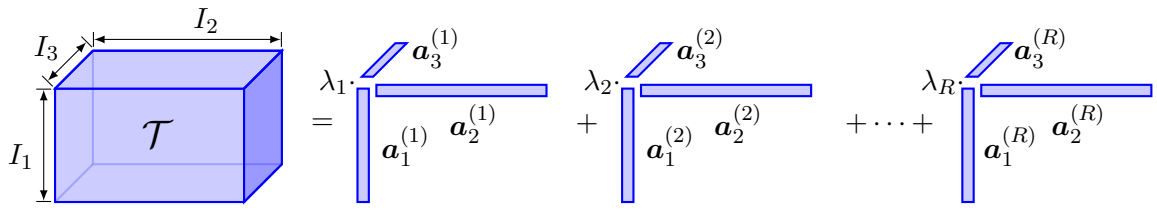


Figure B.1: Décomposition CP d'un tenseur d'ordre 3.

L'objectif de la décomposition CP est de trouver toutes les composantes λ_r et $\boldsymbol{a}_i^{(r)}, 1 \leq i \leq N, 1 \leq r \leq R$, à partir du tenseur $\boldsymbol{\mathcal{T}}$ en utilisant un algorithme comme la méthode des moindres carrées (ALS), par exemple.Cependant, dans la pratique, les tenseurs sont corrompus par du bruit, si bien que nous devons résoudre un problème d'approximation de rang faible, une fois que le rang du tenseur devient inconnue. Ainsi, le problème d'approximation peut être posé de la façon suivante:

$$\underset{\lambda_r, \|\boldsymbol{a}_i^{(r)}\|=1}{\arg\min} \; \|\boldsymbol{\mathcal{T}} - \sum_{r=1}^{R} \lambda_r (\boldsymbol{a}_1^{(r)} \otimes \cdots \otimes \boldsymbol{a}_N^{(r)})\|. \tag{B.2}$$

Cette thèse présente différents objectifs et est divisée en quatre autres chapitres hors cette introduction. Dans le chapitre suivant, notre objective est calculer l'approximation tensorielle de rang-1 en utilisant des méthodes itératifs et non-itératifs. Dans le troisième chapitre, nous utilisons des algorithmes non-itératifs de rang-1 dans le contexte de la déflation itérative, avec le but de résoudre le problème d'approximation (B.2). Le quatrième chapitre établit un lien entre la meilleure approximation de rang-1 et les systèmes quadratiques multivariés. Finalement, nous traitons le cas particulier de la décomposition CP exacte pour des tenseurs avec l'une des matrices facteurs semi-unitaire.

Chapitre: Approximation de Rang-1

Nous étudions des alternatives pour résoudre le problème (B.2) lorsque $R = 1$. En fait, ce problème est NP- difficile, de sorte que de nouveaux algorithmes pour le traiter ont besoin d'être proposés.

Ainsi, nous présentons les principales contributions de ce chapitre:

- *Adaptation du problème de meilleure approximation de rang-1 pour assurer l'estimation de la solution global en utilisant la méthode de Lasserre [Lasserre 2001].* Dans le problème

$$p^{\min} = \min_{\lambda, \boldsymbol{a}_i \in \mathbb{K}^{I_i}} \|\boldsymbol{\mathcal{T}} - \lambda \cdot \boldsymbol{a}_1 \otimes \cdots \otimes \boldsymbol{a}_N\|^2 \quad s.t. \ \|\boldsymbol{a}_i\| = 1. \quad , \tag{B.3}$$

  nous établissons une valeur maximale pour λ afin d'assurer l'existence du minimum global des problèmes hiérarchiques de Lasserre (voir l'appendice A.2 du manuscrit en anglais).

- *Algorithme non-itératif pour calculer une approximation de rang-1.* Nous proposons un algorithme qui s'appelle SeROAP, dont l'idée est basée sur des successives SVDs et projections. Comparé à la méthode de HOSVD tronquée (T-HOSVD) [De Lathauwer 2000], l'algorithme SeROAP présente une complexité numérique plus petite. Par rapport à l'algorithme ST-HOSVD [Vannieuwenhoven 2012], SeROAP est un algorithme compétitif en termes de complexité numérique pour des tenseurs de grandes dimensions et petit ordre.

- *Erreur d'approximation de SeROAP*. Nous démontrons algébriquement que l'erreur d'approximation obtenue avec SeROAP est au moins aussi bon que celle calculée avec l'algorithme ST-HOSVD (et par conséquent que HOSVD tronquée), pour des tenseurs d'ordre 3.

- *Limitation de SeROAP*. Les résultats numériques ont montré que l'algorithme SeROAP présente une performance faible de l'erreur d'approximation pour des tenseurs d'ordre supérieur à quatre, par rapport aux autres méthodes non-itératifs de rang-1 déjà mentionnées. En fait, même pour l'ordre 4, la performance de ST-HOSVD sous ce critère surpasse celle de notre algorithme. En revanche, SeROAP est statistiquement mieux que la HOSVD tronquée avec une probabilité élevée pour cette même ordre.

- *Algorithme d'approximation de rang-1 itératif.* Nous proposons un algorithme appelé CE dont l'idée est construite à partir d'un problème de valeur propre couplée. L'objectif de cet algorithme est de résoudre le problème d'approximation de rang-1 pour le cas d'ordre 3. Nous décrivons également des résultats théoriques assurant l'amélioration et la convergence de l'objectif du problème (B.3).

- *Performance de l'algorithme CE*. La méthode CE présente d'avantage par rapport à ALS. D'abord, en termes de complexité numériques, CE est plus attractive si l'une des dimensions du tenseur est largement supérieur aux autres dimensions. En plus, nous montrons par des simulations que, pour deux types d'initialisations, l'erreur d'approximation obtenue par CE est plus petite que celle calculée par ALS. De même pour le nombre moyen d'itération pour la convergence.

Chapitre: Déflation Itérative

Ce chapitre présente un étude détaillé sur l'utilisation d'algorithmes d'approximation de rang-1 dans le contexte de la déflation itérative (appelé DCPD dans notre manuscrit). L'objective est de calculer une approximation de rang faible pour un tenseur donné.

La déflation itérative combinée avec la meilleure approximation de rang-1 est illustrée dans la Figure B.2.

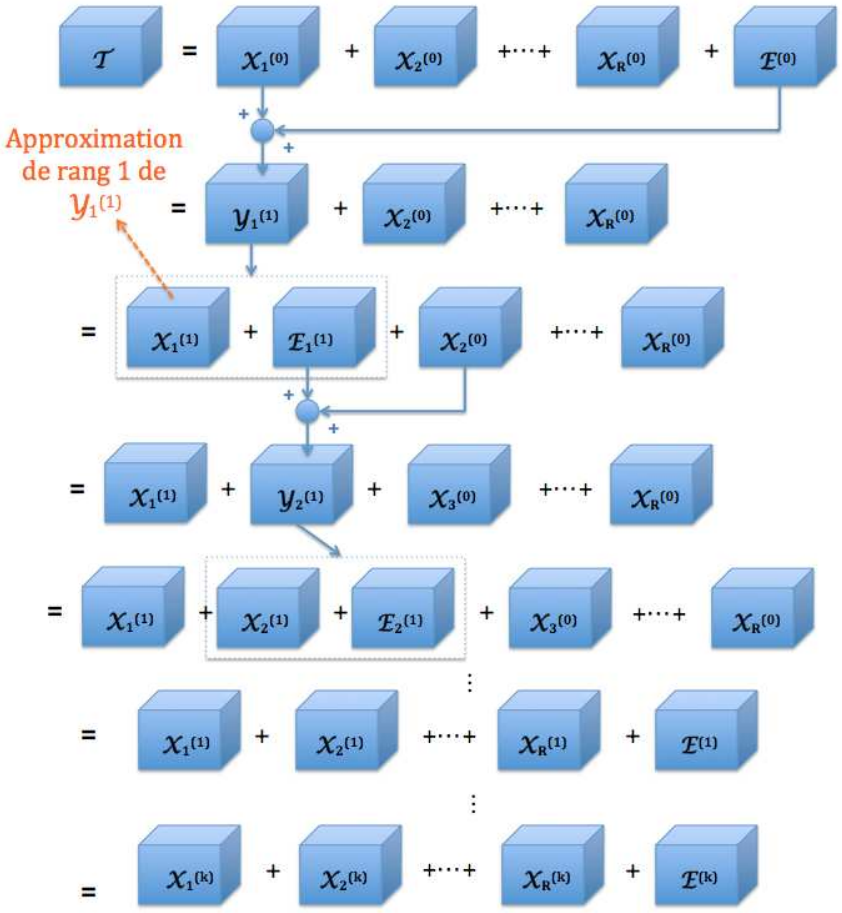


Figure B.2: Déflation itérative + meilleure approximation de rang-1.

À l'itération [1] zéro, nous calculons les tenseurs $\boldsymbol{\mathcal{X}}_r^{(0)}$, $1 < r \leq R$, comme les meilleures approximations de rang-1 des respectifs tenseurs $\boldsymbol{\mathcal{T}} - \sum_{j=1}^{r-1} \boldsymbol{\mathcal{X}}_j^{(0)}$. Pour $r = 1$, $\boldsymbol{\mathcal{X}}_1^{(0)}$ est la meilleure approximation de rang-1 de $\boldsymbol{\mathcal{T}}$. Une fois que cette procédure de déflation ne réduit pas le rang en général [Stegeman 2007], il existe un résidu $\boldsymbol{\mathcal{E}}^{(0)}$ tel que $\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{E}}^{(0)} + \sum_{r=1}^{R} \boldsymbol{\mathcal{X}}_r^{(0)}$.

À la itération 1, on combine le résidu $\boldsymbol{\mathcal{E}}^{(0)}$ avec la première composante $\boldsymbol{\mathcal{X}}_1^{(0)}$ de façon à obtenir le tenseur $\boldsymbol{\mathcal{Y}}_1^{(1)} = \boldsymbol{\mathcal{E}}^{(0)} + \boldsymbol{\mathcal{X}}_1^{(0)}$. Alors, nous décomposons $\boldsymbol{\mathcal{Y}}_1^{(1)}$ comme la somme de sa meilleure approximation de rang-1, i.e. $\boldsymbol{\mathcal{X}}_1^{(1)}$, et un résidu $\boldsymbol{\mathcal{E}}_1^{(1)}$. Alors, la première composante de rang-1

[1] Dans la figure, l'itération est représentée par l'exposant entre parenthèses.

du tenseur $\mathcal{T}$ est mise à jour. Ensuite, on combine le résidu $\boldsymbol{\mathcal{E}}_1^{(1)}$ avec la deuxième composante $\boldsymbol{\mathcal{X}}_2^{(0)}$ pour construire le tenseur $\boldsymbol{\mathcal{Y}}_2^{(1)}$, lequel est décomposé de la même manière que $\boldsymbol{\mathcal{Y}}_1^{(1)}$, Cela nous permet de mettre à jour la deuxième composante appelée $\boldsymbol{\mathcal{X}}_2^{(1)}$. Nous continuons avec la même procédure jusqu'à la composante R, de manière à obtenir un nouveau résidu $\boldsymbol{\mathcal{E}}^{(1)}$ tel que $\|\boldsymbol{\mathcal{E}}^{(1)}\| \leq \|\boldsymbol{\mathcal{E}}^{(0)}\|$. La déflation itérative continue jusqu'à la convergence (ou un autre critère d'arrêt), représentée dans la figure par l'itération k.

Les contributions de ce chapitre sont les suivantes:

- *Déflation itérative + algorithme itératif d'approximation de rang-1.* Avec le but de réduire le plus rapidement possible la norme des résidus $\|\boldsymbol{\mathcal{E}}^{(k)}\|$, on introduit les algorithmes SeROAP, ST-HOSVD et T-HOSVD pour mettre à jour les composantes $\boldsymbol{\mathcal{X}}_r^{(j)}$, ce qui entraîne plus d'intelligence dans la procédure de mise à jour des composantes en comparaison à la méthode de déflation itérative HALS [Cichocki 2009c].
- *Étude théorique sur la convergence de l'algorithme DCPD.* Nous montrons que la norme des résidus $\|\boldsymbol{\mathcal{E}}^{(k)}\|,\ k \geq 1$, est une séquence décroissante. Nous montrons aussi que l'algorithme DCPD estime la décomposition CP exacte d'un tenseur donné lorsque les résidus n'appartient pas à un cône de volume arbitrairement petit. Dans une seconde étape, sous les contraintes de probabilité absolument continue de la distribution des tenseurs et de la continuité d'une certaine fonction des résidus, on montre que DCPD + meilleure approximation de rang-1 peut réduire le résidu initial d'un facteur inférieur à une constante connue (qui dépend de l'angle du cône) après L itérations avec une probabilité élevée. Finalement, on conjecture (Conjecture 3.2.8) l'existence des mesures de probabilités qui assurent la convergence de DCPD + meilleure approximation de rang-1 vers la décomposition exacte avec une haute probabilité.
- *Performance de DCPD avec les algorithmes d'approximation de rang-1 non-itératifs.* Notre conjecture est renforcée par des résultats numériques. On génère des tenseurs dont ces composantes suivent une distribution uniforme. Nous montrons également la réduction rapide du résidu en utilisant DCPD + SeROAP par rapport à la méthode de déflation itérative HALS. Nous présentons aussi un ensemble de résultats numériques pour des scénarios bruités et non-bruités, avec la variation du rang et de l'ordre tensoriels.

Chapitre: Systèmes Quadratiques Multivariés et la Meilleure Approximation de Rang-1

Dans ce chapitre, l'objectif est de montrer que le système quadratique

$$\textbf{P1:}\ \begin{cases} \boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{b}_1^{\mathsf{T}}\boldsymbol{x} + c_1 = 0 \\ \qquad\qquad \vdots \\ \boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}_m\boldsymbol{x} + \boldsymbol{b}_m^{\mathsf{T}}\boldsymbol{x} + c_m = 0, \end{cases}$$

où $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{A}_j \in \mathbb{R}^{n\times n}$ sont des matrices symétriques, $\boldsymbol{b}_j \in \mathbb{R}^n$ et $c_j \in \mathbb{R}$, $1 \leq j \leq m$, peut être réduit à un problème de meilleure approximation de rang-1 d'un tenseur d'ordre 3. La généralisation pour la cas complexe est faite à partir du système P2 ci-après

$$\textbf{P2:} \begin{cases} \boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}_1\boldsymbol{x} + \boldsymbol{b}_1^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{c}_1^{\mathsf{T}}\boldsymbol{x}^* + d_1 = 0 \\ \qquad\qquad \vdots \\ \boldsymbol{x}^{\mathsf{H}}\boldsymbol{A}_m\boldsymbol{x} + \boldsymbol{b}_m^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{c}_m^{\mathsf{T}}\boldsymbol{x}^* + d_m = 0 \end{cases},$$

où $\boldsymbol{x} \in \mathbb{C}^n$, $\boldsymbol{A}_j \in \mathbb{C}^{n\times n}$, $\boldsymbol{b}_j \in \mathbb{C}^n$, $\boldsymbol{c}_j \in \mathbb{C}^n$, et $d_j \in \mathbb{C}$, pour $j = 1, \ldots, m$. Dans ce cas, nous montrons que P2 peut être réduit à un problème de meilleure approximation de rang-1 conjugué et partialement symétrique, lequel est représenté par le problème suivant

$$\textbf{P3:} \begin{cases} p_3^\star = \min\limits_{\boldsymbol{y},\boldsymbol{w}} \ \|\boldsymbol{\mathcal{T}} - \boldsymbol{y}\otimes\boldsymbol{y}^*\otimes\boldsymbol{w}\|. \\ \\ s.t.\ \|\boldsymbol{y}\| = 1, \end{cases}$$

avec $\boldsymbol{\mathcal{T}} \in \mathbb{C}^{n+1\times n+1\times \tilde{K}}$, et $\tilde{K} \leq \operatorname{rank}\{\tilde{\boldsymbol{M}}\}$, où $\tilde{\boldsymbol{M}}$ est une matrice semi-définie positive construite à partir des paramètres du système P2 et $\boldsymbol{y} = \alpha[\boldsymbol{x}\ \ 1]$, pour $\alpha \in \mathbb{C}$.

Les contributions du chapitre sont les suivantes:

- *Réduction du système quadratique au problème de meilleure approximation de rang-1.* Nous montrons qu'une solution d'un système polynomial quadratique dans le corps des réels peut être obtenue à partir de la solution du problème (B.3) pour le cas d'ordre 3. De la même façon, on démontre que le cas le plus général P2 est réduit au problème P3. Ainsi, nous pouvons appliquer un nouvel approche pour traiter des systèmes quadratiques multivariés.
- *Algorithme pour l'approximation de rang-1.* Nous proposons un algorithme pour résoudre le problème de meilleure approximation de rang-1 conjugué et partiellement symétrique. Cet algorithme s'appelle CPS (Conjugated Partially Symmetric) et son idée est basée sur les équations des points stationnaires du problème P3.
- *Performance de l'approche de rang-1.* Les algorithmes itératifs ALS et CPS présentent des performances satisfaisantes lors de l'extraction d'une solution des systèmes quadratiques et dans un temps de simulation raisonnable. Nous présentons quelques exemples des systèmes et aussi un ensemble des simulations pour prouver l'efficacité de l'approche dans les cas réel et complexe.

Chapitre: Projection Alternée Appliquée aux Tenseurs Orthogonaux

L'objectif de ce chapitre est de résoudre la décomposition CP exacte dans le cas où une des matrices facteurs est semi-unitaire. Pour cela, on combine la méthode de déflation avec celle de projection alternée. Les concepts des transversalité et non-tangentialité sont également étudiés dans le contexte tensoriel pour évaluer la convergence d'un algorithme. Les principaux résultats sont les suivants:

- *Algorithme CAPD.* Nous proposons un algorithme appelé CAPD que combine la méthode des projections alternées avec la déflation de tenseurs de rang-1. Cela nous permet de résoudre la décomposition CP exacte d'un tenseur d'ordre 3 avec l'une des matrices facteurs semi-unitaire.

- *Extraction des bonnes composantes de rang-1.* Dans le cas de convergence de l'étape de projection alternée de l'algorithme CAPD, nous montrons qu'une correcte composante de rang-1 du tenseur est toujours obtenue.

- *Convergence vs. transversalité.* Sous des certaines conditions, nous montrons que s'il existe des points transverses à l'intersection des variétés qui définissent la décomposition CP orthogonal de facteur semi-unitaire. Ainsi, pour une initialisation de l'algorithme CAPD proche des variétés impliquées, la transversalité assure une convergence linéaire [Lewis 2008] vers un point à l'intersection des variétés. Cependant, le concept de transversalité est limité dans notre cas, une fois que l'estimation correcte de seulement quelques composantes de rang-1 du tenseur peut être assurée.

- *Convergence vs. non-tangentialité.* Nous démontrons un résultat plus fort basé sur le concept de la non-tangentialité entre variétés. Pour l'algorithme CAPD, nous prouvons que l'estimation de toutes les composantes de rang-1 peuvent être assurée lorsque des points à l'intersection des variétés impliquées sont non-tangentielles. Effectivement, le concept de tranversalité est plus restreint que celui de la non-tangetialité [Andersson 2013].

- *Performance de l'algorithme CAPD.* Nos présentons des résultats numériques montrant que la distance entre les estimations à chaque itération de l'algorithme CAPD converge log-linéairement. Nous montrons aussi la problématique existante lors de l'estimation de toutes les matrices facteurs à la fois, avec la méthode de projection alternée. En fait, sans la déflation de tenseurs de rang-1, l'estimation de colonnes proportionnelles dans le matrices facteurs est possible, de façons que n'existe aucune assurance de trouver les facteurs correctement. Les résultats numériques ont montré que cela est un problème courant.

Perspectives

Nous présentons maintenant les principales perspectives de mon travail de thèse. Les détails des activités envisagées sont décrits par chapitre.

Chapitre: Approximation de Rang-1

- Développer d'autres méthodes dans la Géométrie Algébrique pour résoudre le problème de meilleure approximation de rang-1. Malgré les techniques décrites en [Bucero 2014, Nie 2014] présentent une meilleure performance que la méthode de Lasserre en termes de complexité numérique, elles subissent à une augmentation exponentielle du nombres de variables lors de la relaxation des problèmes d'optimisation. Ainsi, une fois que le nombre de monômes de l'objectif du problème (B.3) est parcimonieux, nous espérons réduire la complexité en tenant compte de ce fait.

- Nous envisageons soit une amélioration de SeROAP, soit la proposition d'un nouvel algorithme non-itératif de rang-1 pour traiter des tenseurs d'ordre supérieur, afin de réduire l'erreur d'approximation. Dans le cas de SeROAP, une idée d'amélioration c'est de choisir l'ordre de dépliage par rapport le rang des versions dépliées du tenseur. Plus précisément, on choisit l'ordre

de dépliage dont les matrices aient le rang le plus petit. Ainsi, on garde plus d'information du tenseur déplié.

- Pour algorithme CE, nous n'avons pas encore prouvé sa convergence vers un point stationnaire, mais seulement que l'objectif du problème de meilleure approximation de rang-1 converge vers une valeur limite. La démonstration de convergence vers la solution optimale est encore un défi.

Chapitre: Déflation Itérative

- La preuve de la Conjecture 3.2.8 est un problème ouvert, même pour des cas particulières (e.g. tenseurs orthogonaux). Nous envisageons d'utiliser les outils mathématiques de la théorie de la mesure et de la géométrie algébrique pour traiter ce problème.

Chapitre: Systèmes Quadratiques Multivariés et la Meilleure Approximation de Rang-1

- Il y a des questions autour de la réduction des systèmes quadratiques à des problèmes d'approximation de rang-1 pour lesquelles nous n'avons pas encore de réponses. Par exemple: quelles sont les conditions assurant l'équivalence entre les problèmes P1 et la meilleure approximation de rang-1? Nous envisageons de traiter cette question bientôt.

Chapitre: Projection Alternée Appliquée aux Tenseurs Orthogonaux

- Nous n'avons pas encore évalué la performance de l'algorithme CAPD sous des scénarios bruités. Les résultats théoriques marchent seulement pour le cas de la décomposition CP exacte. Cependant, nous avons l'intention d'évaluer l'impact du bruit sur l'estimation des matrices facteurs. Une idée pour mitiger des erreurs d'estimation est d'approximer les formes dépliées des tenseurs à des matrices de rang désiré, avant la déflation. D'ailleurs, avec cette approche, nous ne savons pas si l'estimation des facteurs sera satisfaisante comparée à d'autres méthodes telles que celles décrites en [Sørensen 2012].

Bibliography

[Abo 2009] Hirotachi Abo, Giorgio Ottaviani and Chris Peterson. *Induction for secant varieties of Segre varieties*. Transactions of the American Mathematical Society, vol. 361, no. 2, pages 767–792, 2009. (Cited in page(s) 24.)

[Andersson 2013] Fredrik Andersson and Marcus Carlsson. *Alternating projections on nontangential manifolds*. Constructive Approximation, vol. 38, no. 3, pages 489–525, 2013. (Cited in page(s) 85, 88, 93, 95, 96 and 117.)

[Bauschke 2002] Heinz H Bauschke, Patrick L Combettes and D Russell Luke. *Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization*. JOSA A, vol. 19, no. 7, pages 1334–1345, 2002. (Cited in page(s) 88.)

[Becker 2014] H. Becker, L. Albera, P. Comon, M. Haardt, G. Birot, F. Wendling, M. Gavaret, C.-G. Bénar and I. Merlet. *EEG extended source localization: tensor-based vs. conventional methods*. NeuroImage, vol. 96, pages 143–157, 2014. (Cited in page(s) 15.)

[Berg 1987] Christian Berg. *The multidimensional moment problem and semigroups*. In Proc. Symp. Appl. Math, volume 37, pages 110–124, 1987. (Cited in page(s) 108.)

[Brachat 2009] Jerome Brachat, Pierre Comon, Bernard Mourrain and Elias Tsigaridas. *Symmetric tensor decomposition*. In Signal Processing Conference, 2009 17th European, pages 525–529. IEEE, 2009. (Cited in page(s) 68.)

[Breg 1965] LM Breg. *Finding the common point of convex sets by the method of sucessive projections*. Dokl. Akad. Mousk SSSR, pages 487–490, 1965. (Cited in page(s) 88.)

[Bucero 2014] M. A. Bucero and B. Mourrain. *Border Basis relaxation for polynomial optimization*. arXiv preprint arXiv:1404.5489, 2014. (Cited in page(s) 16, 24, 25, 27, 46, 68, 104 and 117.)

[Cichocki 2009a] A. Cichocki and A.-H. Phan. *Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations*. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E92-A, no. 3, pages 708–721, 2009. (Cited in page(s) 16.)

[Cichocki 2009b] Andrzej Cichocki and PHAN Anh-Huy. *Fast local algorithms for large scale nonnegative matrix and tensor factorizations*. IEICE transactions on fundamentals of electronics, communications and computer sciences, vol. 92, no. 3, pages 708–721, 2009. (Cited in page(s) 51.)

[Cichocki 2009c] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan and Shun-ichi Amari. Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons, 2009. (Cited in page(s) 19, 51, 57, 103, 107 and 115.)

[Combettes 1990] PL Combettes and Henry J Trussell. *Method of successive projections for finding a common point of sets in metric spaces*. Journal of optimization theory and applications, vol. 67, no. 3, pages 487–507, 1990. (Cited in page(s) 88.)

[Comon 1990] P. Comon and G. H. Golub. *Tracking a few extreme singular values and vectors in signal processing*. Proceedings of the IEEE, vol. 78, no. 8, pages 1327–1343, 1990. (Cited in page(s) 29 and 50.)

[Comon 2009a] P. Comon, X. Luciani and A. L. F. de Almeida. *Tensor Decompositions, Alternating Least Squares and other Tales*. Jour. Chemometrics, vol. 23, no. 7-8, pages 393–405, Aot 2009. (Cited in page(s) 16.)

[Comon 2009b] P. Comon, X. Luciani and A. L. F. de Almeida. *Tensor decompositions, alternating least squares and other tales*. Journal of Chemometrics, vol. 23, no. 7-8, pages 393–405, 2009. (Cited in page(s) 65, 68 and 107.)

[Comon 2010] P. Comon and C. Jutten, editors. Handbook of blind source separation, independent component analysis and applications. Academic Press, Oxford UK, Burlington USA, 2010. ISBN: 978-0-12-374726-6, hal-00460653. (Cited in page(s) 15.)

[Courtois 2000] Nicolas Courtois, Alexander Klimov, Jacques Patarin and Adi Shamir. *Efficient algorithms for solving overdefined systems of multivariate polynomial equations*. In Advances in CryptologyEUROCRYPT 2000, pages 392–407. Springer, 2000. (Cited in page(s) 68.)

[Cox 1992] David Cox, John Little and Donal O'shea. Ideals, varieties, and algorithms, volume 3. Springer, 1992. (Cited in page(s) 27 and 108.)

[Cox 2006] David A Cox, John Little and Donal O'shea. Using algebraic geometry, volume 185. Springer Science & Business Media Business Media, 2006. (Cited in page(s) 68 and 74.)

[Curto 1991] Raúl E Curto and Lawrence A Fialkow. *Recursiveness, positivity, and truncated moment problems*. Houston Journal of Mathematics, vol. 17, no. 4, pages 603–635, 1991. (Cited in page(s) 108.)

[da Silva 2014] A. P. da Silva, P. Comon and de Almeida A. L. F. *Rank-1 Tensor Approximation Methods and Application to Deflation*. arXiv:1508.05273, 08 2014. (Cited in page(s) 31 and 68.)

[da Silva 2015] A. P. da Silva, P. Comon and A. L. F. de Almeida. *An Iterative Deflation Algorithm for Exact CP Tensor Decomposition*. IEEE conference on Acoustics, Speech and Signal Processing, 2015. (Cited in page(s) 51 and 54.)

[Davenport 2016] Mark Davenport and Justin Romberg. *An overview of low-rank matrix recovery from incomplete observations*. 2016. (Cited in page(s) 68.)

[de Almeida 2006] André LF de Almeida, Gérard Favier, João Cesar M Mota and Raul L De Lacerda. *Estimation of frequency-selective block-fading MIMO channels using PARAFAC modeling and alternating least squares*. In Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on, pages 1630–1634. IEEE, 2006. (Cited in page(s) 17 and 18.)

[de Almeida 2007] A. L. F. de Almeida, G. Favier and J. C. M. Mota. *Parafac-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization*. Signal Processing, vol. 87, no. 2, pages 337–351, Fvrier 2007. (Cited in page(s) 15.)

[De Lathauwer 2000] L. De Lathauwer, B. De Moor and J. Vandewalle. *A multilinear singular value decomposition*. SIAM journal on Matrix Analysis and Applications, vol. 21, no. 4, pages 1253–1278, 2000. (Cited in page(s) 24, 28 and 113.)

[De Lathauwer 2004] Lieven De Lathauwer and Joos Vandewalle. *Dimensionality reduction in higher-order signal processing and rank-(R 1, R 2, ..., R N) reduction in multilinear algebra*. Linear Algebra and its Applications, vol. 391, pages 31–55, 2004. (Cited in page(s) 28.)

[De Lathauwer 2006] L. De Lathauwer. *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*. SIAM Journal on Matrix Analysis and Applications, vol. 28, no. 3, pages 642–666, 2006. (Cited in page(s) 17.)

[De Lathauwer 2007] Lieven De Lathauwer, Joséphine Castaing and Jean-François Cardoso. *Fourth-order cumulant-based blind identification of underdetermined mixtures*. Signal Processing, IEEE Transactions on, vol. 55, no. 6, pages 2965–2973, 2007. (Cited in page(s) 89.)

[De Silva 2008] V. De Silva and L.-H. Lim. *Tensor rank and the ill-posedness of the best low-rank approximation problem*. SIAM Journal on Matrix Analysis and Applications, vol. 30, no. 3, pages 1084–1127, 2008. (Cited in page(s) 16.)

[Dennis Jr 1996] John E Dennis Jr and Robert B Schnabel. Numerical methods for unconstrained optimization and nonlinear equations, volume 16. Siam, 1996. (Cited in page(s) 68 and 82.)

[Friedland 2013] Shmuel Friedland, Volker Mehrmann, Renato Pajarola and SK Suter. *On best rank one approximation of tensors*. Numerical Linear Algebra with Applications, vol. 20, no. 6, pages 942–955, 2013. (Cited in page(s) 24, 68, 72, 73 and 76.)

[Friedland 2014] Shmuel Friedland and Giorgio Ottaviani. *The number of singular vector tuples and uniqueness of best rank-one approximation of tensors*. Foundations of Computational Mathematics, vol. 14, no. 6, pages 1209–1242, 2014. (Cited in page(s) 26.)

[Grellier 1999] Olivier Grellier and Pierre Comon. *Closed-form equalization*. In Signal Processing Advances in Wireless Communications, 1999. SPAWC'99. 1999 2nd IEEE Workshop on, pages 219–222. IEEE, 1999. (Cited in page(s) 24 and 25.)

[Grellier 2000] Olivier Grellier and Pierre Comon. *Analytical blind discrete source separation*. In Signal Processing Conference, 2000 10th European, pages 1–4. IEEE, 2000. (Cited in page(s) 24 and 25.)

[Grigoriadis 1999] Karolos M Grigoriadis and Eric B Beran. Alternating projection algorithms for linear matrix inequalities problems with rank constraints. Society for Industrial and Applied Mathematics, 1999. (Cited in page(s) 88.)

[Grigoriev 2005] Dima Grigoriev and Dmitrii V Pasechnik. *Polynomial-time computing over quadratic maps i: sampling in real algebraic sets*. computational complexity, vol. 14, no. 1, pages 20–52, 2005. (Cited in page(s) 68 and 74.)

[Gubin 1967] LG Gubin, BT Polyak and EV Raik. *The method of projections for finding the common point of convex sets*. USSR Computational Mathematics and Mathematical Physics, vol. 7, no. 6, pages 1–24, 1967. (Cited in page(s) 88.)

[Hackbusch 2012] Wolfgang Hackbusch. Tensor spaces and numerical tensor calculus, volume 42. Springer Science & Business Media, 2012. (Cited in page(s) 25.)

[Henrion 2003] Didier Henrion and Jean-Bernard Lasserre. *GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi*. ACM Transactions on Mathematical Software (TOMS), vol. 29, no. 2, pages 165–194, 2003. (Cited in page(s) 27 and 110.)

[Henrion 2005] Didier Henrion and Jean-Bernard Lasserre. *Detecting global optimality and extracting solutions in GloptiPoly*. In Positive polynomials in control, pages 293–310. Springer, 2005. (Cited in page(s) 110.)

[Henrion 2009] Didier Henrion, Jean-Bernard Lasserre and Johan Löfberg. *GloptiPoly 3: moments, optimization and semidefinite programming*. Optimization Methods & Software, vol. 24, no. 4-5, pages 761–779, 2009. (Cited in page(s) 27 and 110.)

[Higham 2002] Nicholas J Higham. *Computing the nearest correlation matrix - a problem from finance*. IMA journal of Numerical Analysis, vol. 22, no. 3, pages 329–343, 2002. (Cited in page(s) 88.)

[Hillar 2013] C. J. Hillar and L.-H. Lim. *Most tensor problems are NP-hard*. Journal of the ACM (JACM), vol. 60, no. 6, page 45, 2013. (Cited in page(s) 16.)

[Hitchcock 1927] F. L. Hitchcock. *The Expression of a Tensor or a Polyadic as a Sum of Products*. Journal of Mathematics and Physics, vol. 6, pages 164–189, 1927. (Cited in page(s) 15 and 112.)

[Kolda 2001] Tamara G Kolda. *Orthogonal tensor decompositions*. SIAM Journal on Matrix Analysis and Applications, vol. 23, no. 1, pages 243–255, 2001. (Cited in page(s) 89.)

[Kolda 2009] T. G. Kolda and B. W. Bader. *Tensor decompositions and applications*. SIAM review, vol. 51, no. 3, pages 455–500, 2009. (Cited in page(s) 15, 29, 56 and 71.)

[Kruskal 1977] Joseph B Kruskal. *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*. Linear algebra and its applications, vol. 18, no. 2, pages 95–138, 1977. (Cited in page(s) 17.)

[Landsberg 2012] Joseph M Landsberg. Tensors: geometry and applications. American Mathematical Society, 2012. (Cited in page(s) 24.)

[Lasserre 2001] J. B. Lasserre. *Global optimization with polynomials and the problem of moments*. SIAM Journal on Optimization, vol. 11, no. 3, pages 796–817, 2001. (Cited in page(s) 16, 24, 25, 68, 108, 109 and 113.)

[Lathauwer 2000] L. De Lathauwer, B. De Moor and J. Vandewalle. *On the best rank-1 and rank-(R1,R2,. . . RN) approximation of High-Order Tensors*. SIAM Jour. Matrix Ana. Appl., vol. 21, no. 4, pages 1324–1342, Avril 2000. (Cited in page(s) 68.)

[Laurent 2009] Monique Laurent. *Sums of squares, moment matrices and optimization over polynomials*. In Emerging applications of algebraic geometry, pages 157–270. Springer, 2009. (Cited in page(s) 25, 109 and 110.)

[Lebrun 2004] Jérôme Lebrun and Ivan Selesnick. *Gröbner bases and wavelet design*. Journal of Symbolic Computation, vol. 37, no. 2, pages 227–259, 2004. (Cited in page(s) 68.)

[Lewis 2008] Adrian S Lewis and Jérôme Malick. *Alternating projections on manifolds*. Mathematics of Operations Research, vol. 33, no. 1, pages 216–234, 2008. (Cited in page(s) 85, 88, 91, 93, 94 and 117.)

[Li 1997] Tien-Yien Li. *Numerical solution of multivariate polynomial systems by homotopy continuation methods*. Acta numerica, vol. 6, pages 399–436, 1997. (Cited in page(s) 68.)

[Lim 2014] Lek-Heng Lim and Pierre Comon. *Blind multilinear identification*. Information Theory, IEEE Transactions on, vol. 60, no. 2, pages 1260–1280, 2014. (Cited in page(s) 17.)

[Lin 2004] Zhiping Lin, Li Xu and Qinghe Wu. *Applications of Gröbner bases to signal and image processing: A survey*. Linear algebra and its applications, vol. 391, pages 169–202, 2004. (Cited in page(s) 68.)

[Lipton 2004] Richard J Lipton and Evangelos Markakis. *Nash equilibria via polynomial equations*. In LATIN 2004: Theoretical Informatics, pages 413–422. Springer, 2004. (Cited in page(s) 68.)

[Liu 2001] Xiangqian Liu and Nicholas D Sidiropoulos. *Cramér-Rao lower bounds for low-rank decomposition of multidimensional arrays*. Signal Processing, IEEE Transactions on, vol. 49, no. 9, pages 2074–2086, 2001. (Cited in page(s) 17.)

[Miron 2008] Sebastian Miron, Xijing Guo and David Brie. *DOA estimation for polarized sources on a vector-sensor array by PARAFAC decomposition of the fourth-order covariance tensor*. In Signal Processing Conference, 2008 16th European, pages 1–5. IEEE, 2008. (Cited in page(s) 89.)

[Mourrain 2012] Bernard Mourrain and Philippe Trebuchet. *Border basis representation of a general quotient algebra*. In Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation, pages 265–272. ACM, 2012. (Cited in page(s) 28.)

[Muti 2005] Damien Muti and Salah Bourennane. *Multidimensional filtering based on a tensor approach*. Signal Processing, vol. 85, no. 12, pages 2338–2353, 2005. (Cited in page(s) 28.)

[Nicolaides 1987] Roy A Nicolaides. *Deflation of conjugate gradients with applications to boundary value problems*. SIAM Journal on Numerical Analysis, vol. 24, no. 2, pages 355–365, 1987. (Cited in page(s) 50.)

[Nie 2014] J. Nie and L. Wang. *Semidefinite relaxations for best rank-1 tensor approximations*. SIAM Journal on Matrix Analysis and Applications, vol. 35, no. 3, pages 1155–1179, 2014. (Cited in page(s) 24, 27, 28 and 117.)

[Paatero 1997] P. Paatero. *A weighted non-negative least squares algorithm for three-way Parafac factor analysis*. Chemometrics Intell. Lab. Syst., vol. 38, pages 223–242, 1997. (Cited in page(s) 16.)

[Pardalos 1991] Panos M Pardalos and Stephen A Vavasis. *Quadratic programming with one negative eigenvalue is NP-hard*. Journal of Global Optimization, vol. 1, no. 1, pages 15–22, 1991. (Cited in page(s) 68.)

[Parrilo 2003] Pablo A Parrilo. *Semidefinite programming relaxations for semialgebraic problems*. Mathematical programming, vol. 96, no. 2, pages 293–320, 2003. (Cited in page(s) 25 and 68.)

[Phan 2014] A.-H. Phan, P. Tichavsky and A. Cichocki. *Deflation Method for Candecomp/Paarafac tensor decomposition*. In ICASSP, pages 6736–6740, Firenze, Italy, Mai 4-9 2014. (Cited in page(s) 16.)

[Phan 2015a] A.-H. Phan, P. Tichavskỳ and A. Cichocki. *Tensor deflation for CANDECOMP/PARAFAC. Part 1: Alternating Subspace Update Algorithm*. IEEE Transaction on Signal Processing, 2015. (Cited in page(s) 51.)

[Phan 2015b] Anh-Huy Phan, Petr Tichavsky and Andrzej Cichocki. *Tensor deflation for CANDECOMP/PARAFAC. Part 3: Rank splitting*. arXiv preprint arXiv:1506.04971, 2015. (Cited in page(s) 51.)

[Putinar 1999] Mihai Putinar and Florian-Horia Vasilescu. *Solving moment problems by dimensional extension*. Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, vol. 328, no. 6, pages 495–499, 1999. (Cited in page(s) 108.)

[Sahnoun 2015] S. Sahnoun and P. Comon. *Joint Source Estimation and Localization*. IEEE Trans. on Signal Processing, vol. 63, no. 10, 2015. (Cited in page(s) 15.)

[Savas 2003] Berkant Savas. *Analyses and tests of handwritten digit recognition algorithms*. LiTH-MAT-EX-2003-01, Linkping University, Department of Mathematics, 2003. (Cited in page(s) 28.)

[Savas 2007] Berkant Savas and Lars Eldén. *Handwritten digit classification using higher order singular value decomposition*. Pattern recognition, vol. 40, no. 3, pages 993–1003, 2007. (Cited in page(s) 28.)

[Savas 2008] B. Savas. *Algorithms in Data Mining using Matrix and Tensor Metho ds*. PhD thesis, Linköping Univ. Tech., 2008. (Cited in page(s) 15.)

[Schnabel 1984] Robert B Schnabel and Paul D Frank. *Tensor methods for nonlinear equations*. SIAM Journal on Numerical Analysis, vol. 21, no. 5, pages 815–843, 1984. (Cited in page(s) 68 and 74.)

[Shi 2013] Xinchu Shi, Haibin Ling, Junling Xing and Weiming Hu. *Multi-target tracking by rank-1 tensor approximation*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2387–2394, 2013. (Cited in page(s) 24.)

[Sidiropoulos 2000a] Nicholas D Sidiropoulos and Rasmus Bro. *On the uniqueness of multilinear decomposition of N-way arrays*. Journal of chemometrics, vol. 14, no. 3, pages 229–239, 2000. (Cited in page(s) 17.)

[Sidiropoulos 2000b] Nicholas D Sidiropoulos, Georgios B Giannakis and Rasmus Bro. *Blind PARAFAC receivers for DS-CDMA systems*. Signal Processing, IEEE Transactions on, vol. 48, no. 3, pages 810–823, 2000. (Cited in page(s) 17 and 18.)

[Smilde 2005] A. Smilde, R. Bro and P. Geladi. Multi-way analysis: applications in the chemical sciences. John Wiley & Sons, 2005. (Cited in page(s) 15 and 107.)

[Sørensen 2010] Mikael Sørensen, Lieven De Lathauwer and Luc Deneire. *PARAFAC with orthogonality in one mode and applications in DS-CDMA systems*. In Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pages 4142–4145. IEEE, 2010. (Cited in page(s) 89.)

[Sørensen 2012] Mikael Sørensen, Lieven De Lathauwer, Pierre Comon, Sylvie Icart and Luc Deneire. *Canonical polyadic decomposition with a columnwise orthonormal factor matrix*. SIAM Journal on Matrix Analysis and Applications, vol. 33, no. 4, pages 1190–1213, 2012. (Cited in page(s) 89, 101 and 118.)

[Stegeman 2007] Alwin Stegeman and Nicholas D Sidiropoulos. *On Kruskals uniqueness condition for the Candecomp /Parafac decomposition*. Linear Algebra and its applications, vol. 420, no. 2, pages 540–552, 2007. (Cited in page(s) 17 and 114.)

[Stegeman 2010] A. Stegeman and P. Comon. *Subtracting a best rank-1 approximation does not necessarily decrease tensor rank*. Linear Algebra Appl., vol. 433, no. 7, pages 1276–1300, Dcembre 2010. hal-00512275. (Cited in page(s) 50 and 51.)

[Thomae 2012] Enrico Thomae and Christopher Wolf. *Solving underdetermined systems of multivariate quadratic equations revisited*. In Public Key Cryptography–PKC 2012, pages 156–171. Springer, 2012. (Cited in page(s) 68.)

[Uschmajew 2012] A. Uschmajew. *Local convergence of the Alternating Least Squares Algorithm for Canonical Tensor Approximation*. SIAM Jour. matrix Analysis, vol. 33, no. 2, pages 639–652, 2012. (Cited in page(s) 24 and 74.)

[Van Loan 1993] C. F. Van Loan and N. Pitsianis. Approximation with kronecker products. Springer, 1993. (Cited in page(s) 42.)

[Vannieuwenhoven 2012] Nick Vannieuwenhoven, Raf Vandebril and Karl Meerbergen. *A new truncation strategy for the higher-order singular value decomposition*. SIAM Journal on Scientific Computing, vol. 34, no. 2, pages A1027–A1052, 2012. (Cited in page(s) 24, 25, 29, 30, 34, 39 and 113.)

[Vasilescu 2002] M Alex O Vasilescu and Demetri Terzopoulos. *Multilinear analysis of image ensembles: Tensorfaces*. In Computer VisionECCV 2003, pages 447–460. Springer, 2002. (Cited in page(s) 28.)

[Verschelde 1999] Jan Verschelde. *Polynomial homotopies for dense, sparse and determinantal systems*. arXiv preprint math/9907060, 1999. (Cited in page(s) 68, 74 and 82.)

[Von Neumann 1950] John Von Neumann. Functional operators: The geometry of orthogonal spaces. Princeton University Press, 1950. (Cited in page(s) 88.)

[Wang 2014] L. Wang and T. chu. *On the global convergence of the alternating least squares method for rank-one approximation to generic tensors*. SIAM J. Matrix Anal. Appl., vol. 35, pages 1058–1072, 2014. (Cited in page(s) 40.)

[Wang 2015] Liqi Wang, Moody T Chu and Bo Yu. *Orthogonal Low Rank Tensor Approximation: Alternating Least Squares Method and Its Global Convergence*. SIAM Journal on Matrix Analysis and Applications, vol. 36, no. 1, pages 1–19, 2015. (Cited in page(s) 89.)

[Yang 2016] Yuning Yang, Yunlong Feng, Xiaolin Huang and Johan AK Suykens. *Rank-1 Tensor Properties with Applications to a Class of Tensor Optimization Problems*. SIAM Journal on Optimization, vol. 26, no. 1, pages 171–196, 2016. (Cited in page(s) 24.)

[Zangwill 1967] Willard I Zangwill. *Non-linear programming via penalty functions*. Management science, vol. 13, no. 5, pages 344–358, 1967. (Cited in page(s) 88.)

[Zhang 2001] Tong Zhang and Gene H Golub. *Rank-one approximation to high order tensors*. SIAM Journal on Matrix Analysis and Applications, vol. 23, no. 2, pages 534–550, 2001. (Cited in page(s) 24 and 50.)

Abstract: Low rank tensor decomposition has been playing for the last years an important role in many applications such as blind source separation, telecommunications, sensor array processing, neuroscience, chemometrics, and data mining. The Canonical Polyadic tensor decomposition is very attractive when compared to standard matrix-based tools, manly on system identification. In this thesis, we propose: (i) several algorithms to compute specific low rank-approximations: finite/iterative rank-1 approximations, iterative deflation approximations, and orthogonal tensor decompositions. (ii) A new strategy to solve multivariate quadratic systems, where this problem is reduced to a best rank-1 tensor approximation problem. (iii) Theoretical results to study and proof the performance or the convergence of some algorithms. All performances are supported by numerical experiments.

Keywords: CP decomposition, rank-1 approximation, deflation, quadratic system, orthogonal tensor.

Résumé : L'approximation tensorielle de rang faible joue ces dernières années un rôle important dans plusieurs applications, telles que la séparation aveugle de source, les télécommunications, le traitement d'antennes, les neurosciences, la chimiométrie, et l'exploration de données. La décomposition tensorielle Canonique Polyadique est très attractive comparativement à des outils matriciels classiques, notamment pour l'identification de systèmes. Dans cette thèse, nous proposons (i) plusieurs algorithmes pour calculer quelques approximations de rang faible spécifique: approximation de rang-1 itérative et en un nombre fini d'opérations, l'approximation par déflation itérative, et la décomposition tensorielle orthogonale; (ii) une nouvelle stratégie pour résoudre des systèmes quadratiques multivariés, où ce problème peut être réduit à la meilleure approximation de rang-1 d'un tenseur; (iii) des résultats théoriques pour étudier les performances ou prouver la convergence de quelques algorithmes. Toutes les performances sont illustrées par des simulations informatiques.

Mots clés : décomposition CP, approximation de rang-1, déflation, système quadratique, tenseur orthogonal.

Resumo : A aproximação tensorial de baixo posto desempenha nestes últimos anos um papel importante em várias aplicações, tais como separação cega de fontes, telecomunicações, processamento de antenas, neurociênca, quimiometria e exploração de dados. A decomposição tensorial canônica é bastante atrativa se comparada às técnicas matriciais clássicas, principalmente na identificação de sistemas. Nesta tese, propõe-se (i) vários algoritmos para calcular alguns tipos de aproximação de posto: aproximação de posto-1 iterativa e em um número finito de operações, a aproximação por deflação iterativa, e a decomposição tensorial ortogonal; (ii) uma nova estratégia para resolver sistemas quadráticos em várias variáveis, em que tal problema pode ser reduzido à melhor aproximação de posto-1 de um tensor; (iii) resultados teóricos visando estudar o desempenho ou demonstrar a convergência de alguns algoritmos. Todas os desempenhos são ilustrados através de simulações computacionais.

Palavras-chave : decomposição CP, aproximação de posto-1, deflação, sistema quadrático, tensor ortogonal.

GIPSA-lab, 11 rue des Mathématiques, Grenoble Campus BP 46,
F-38402 Saint Martin d'Hères CEDEX